IN THE UNITED STATES DISTRICT COURT

FOR THE EASTERN DISTRICT OF TEXAS

TYLER DIVISION

ALOFT MEDIA, LLC,	§	
	§	
Plaintiff,	§	
	§	**Civil Action No. 6:08-cv-51**
v.	§	
	§	
MICROSOFT CORPORATION,	§	**JURY TRIAL DEMANDED**
	§	
Defendant.	§	
	§	
	§	

EXHIBIT A

Dockets.Justia.com

US006901393B1

(12) United States Patent
Owen et al.

(10) Patent No.: US 6,901,393 B1
(45) Date of Patent: May 31, 2005

(54) SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR A CUSTOMER-CENTRIC COLLABORATIVE PROTOCOL

(75) Inventors: **Daniel L. Owen**, Lakeville, MI (US); **Michael W. Kusnic**, Lakeville, MI (US)

(73) Assignee: **Collaborative Decision Platforms, LLC**, Lakeville, MI (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 582 days.

(21) Appl. No.: **09/708,244**

(22) Filed: **Nov. 7, 2000**

Related U.S. Application Data

(60) Provisional application No. 60/163,984, filed on Nov. 8, 1999.

(51) **Int. Cl.**[7] **G06F 15/18**; G06F 17/60

(52) **U.S. Cl.** **706/46**; 706/45; 706/47; 705/7; 705/10

(58) **Field of Search** 705/46, 7, 10; 706/45, 47, 46

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,517,405 A * 5/1996 McAndrew et al. 706/45
5,732,397 A * 3/1998 DeTore et al. 705/1
5,953,707 A * 9/1999 Huang et al. 705/10
5,995,951 A * 11/1999 Ferguson 706/10
6,012,051 A * 1/2000 Sammon et al. 706/52
6,826,552 B1 * 11/2004 Grosser et al. 706/47

OTHER PUBLICATIONS

Steve Glickman "Interpreting Business Assessment Results" Jun. 1998. pp. 1–4.*

Wong "Preference–Based Decision Making for Cooperative Knowledge–Based Systems" Oct. 1994; Institute for New Generation Computer Technology; pp. 407–435.*

Grier et al "A Decision Support Expert System for Correction of Manufacturing Process Problems"; Dec. 1990; ACM.*

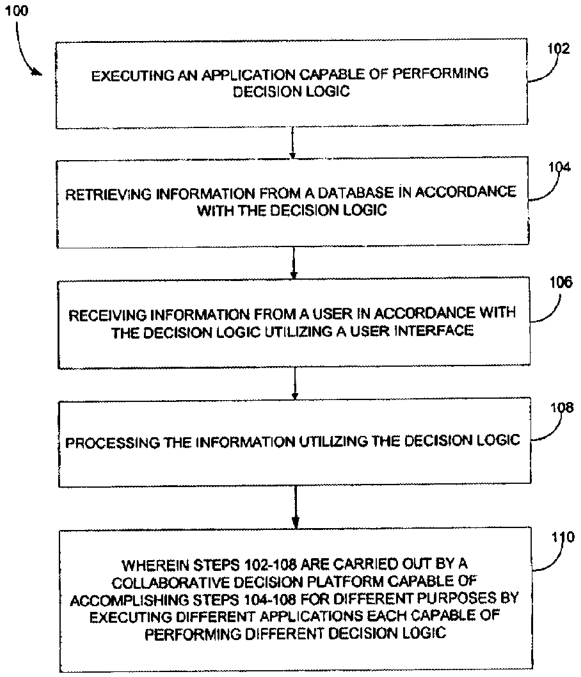
* cited by examiner

Primary Examiner—Romain Jeanty
(74) *Attorney, Agent, or Firm*—Zilka-Kotab, PC

(57) ABSTRACT

A system, method and computer program product are provided for affording customer-centric collaborative decision making in a business-to-business framework. Initially, a minimum set of attributes is defined. Thereafter, first information regarding each of the minimum set of attributes is received from a receiving business. Second information is then received regarding proposed products or services in terms of the minimum set of attributes. Such second information is received from a supplying business. In use, a decision process is executed based on the first information and the second information as to which products or services is suitable for the receiving business.

17 Claims, 49 Drawing Sheets


100
EXECUTING AN APPLICATION CAPABLE OF PERFORMING DECISION LOGIC
102
RETRIEVING INFORMATION FROM A DATABASE IN ACCORDANCE WITH THE DECISION LOGIC
104
RECEIVING INFORMATION FROM A USER IN ACCORDANCE WITH THE DECISION LOGIC UTILIZING A USER INTERFACE
106
PROCESSING THE INFORMATION UTILIZING THE DECISION LOGIC
108
WHEREIN STEPS 102-108 ARE CARRIED OUT BY A COLLABORATIVE DECISION PLATFORM CAPABLE OF ACCOMPLISHING STEPS 104-108 FOR DIFFERENT PURPOSES BY EXECUTING DIFFERENT APPLICATIONS EACH CAPABLE OF PERFORMING DIFFERENT DECISION LOGIC
110

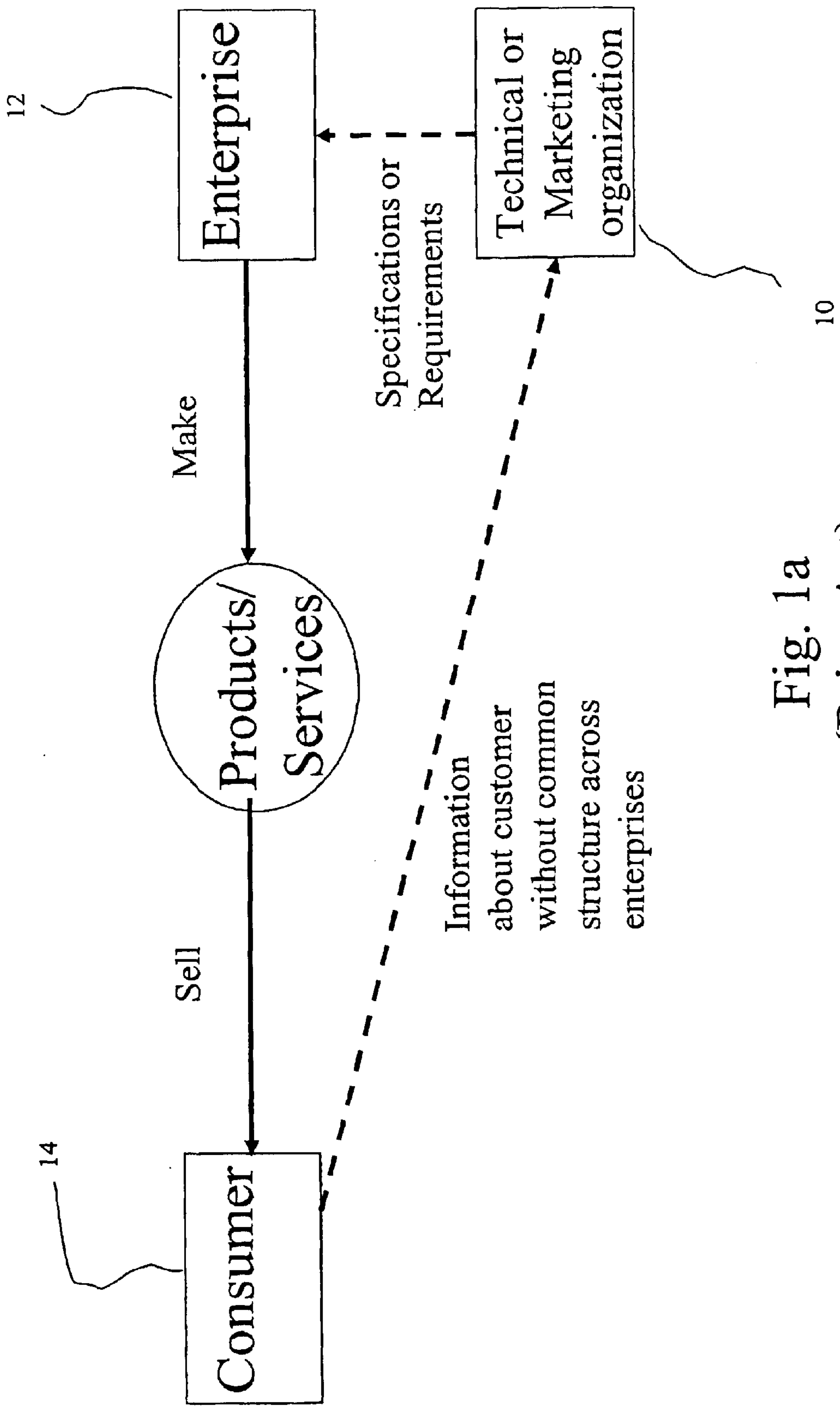

12
Enterprise
Technical or Marketing organization
10
Specifications or Requirements
Make
Products/ Services
Information about customer without common structure across enterprises
Sell
14
Consumer

Fig. 1a (Prior Art)

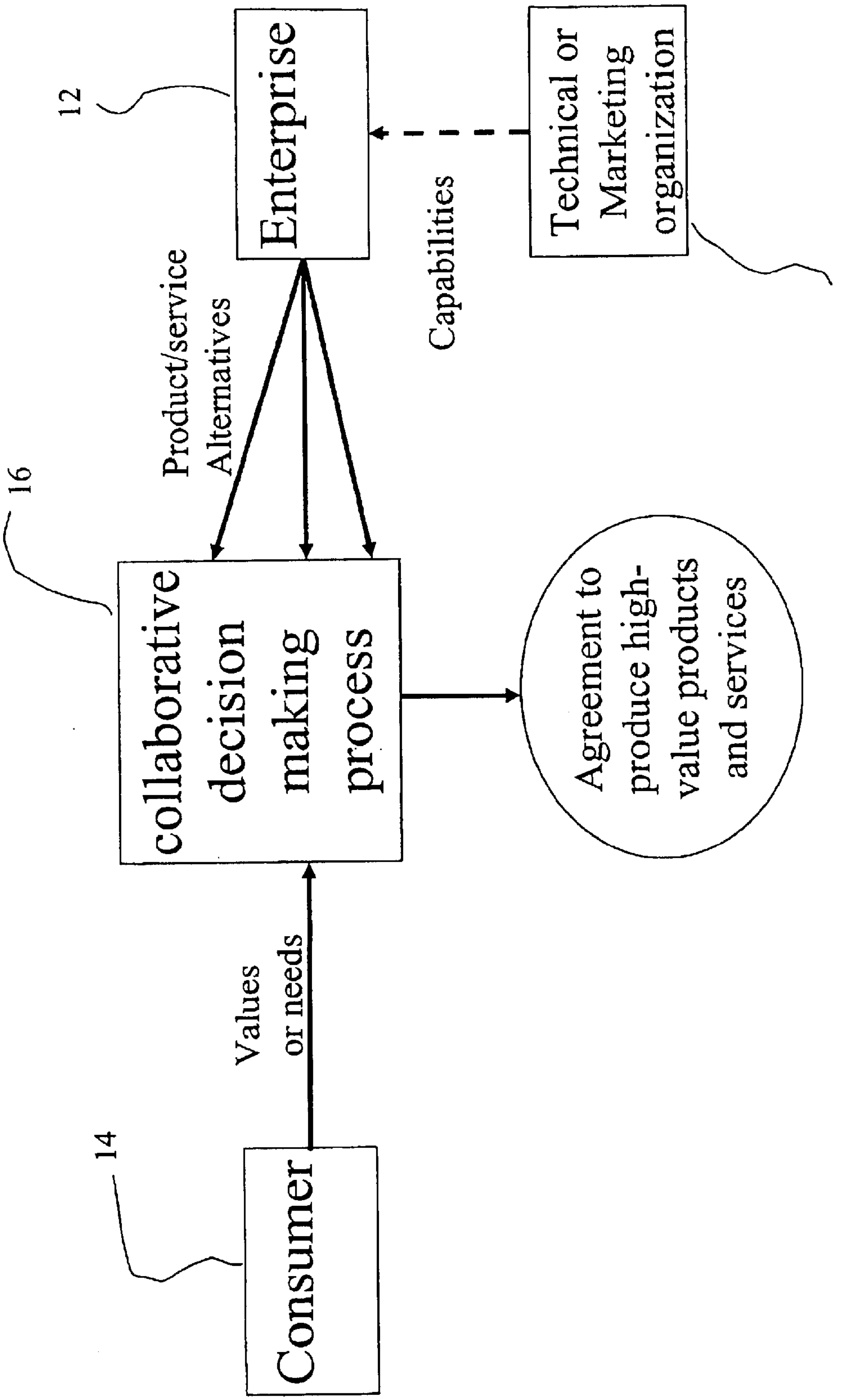

12
Enterprise
Technical or
Marketing
organization
Capabilities
10
Product/service
Alternatives
16
collaborative
decision
making
process
Agreement to
produce high-
value products
and services
14
Consumer
Values
or needs

Fig. 1b
(Prior Art)

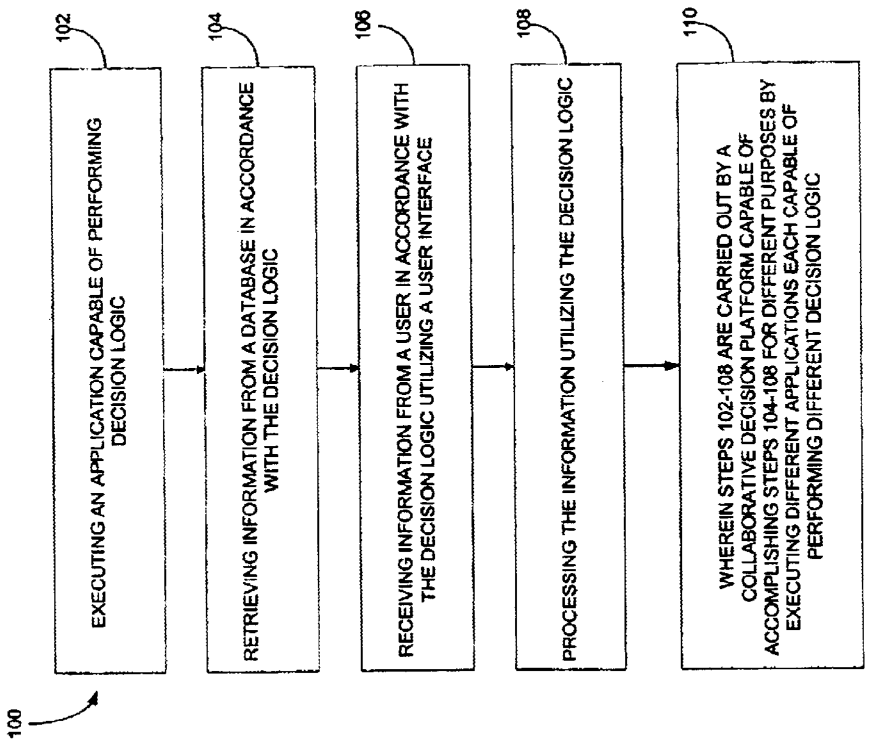

100
102
EXECUTING AN APPLICATION CAPABLE OF PERFORMING DECISION LOGIC
104
RETRIEVING INFORMATION FROM A DATABASE IN ACCORDANCE WITH THE DECISION LOGIC
106
RECEIVING INFORMATION FROM A USER IN ACCORDANCE WITH THE DECISION LOGIC UTILIZING A USER INTERFACE
108
PROCESSING THE INFORMATION UTILIZING THE DECISION LOGIC
110
WHEREIN STEPS 102-108 ARE CARRIED OUT BY A COLLABORATIVE DECISION PLATFORM CAPABLE OF ACCOMPLISHING STEPS 104-108 FOR DIFFERENT PURPOSES BY EXECUTING DIFFERENT APPLICATIONS EACH CAPABLE OF PERFORMING DIFFERENT DECISION LOGIC

Fig. 1c

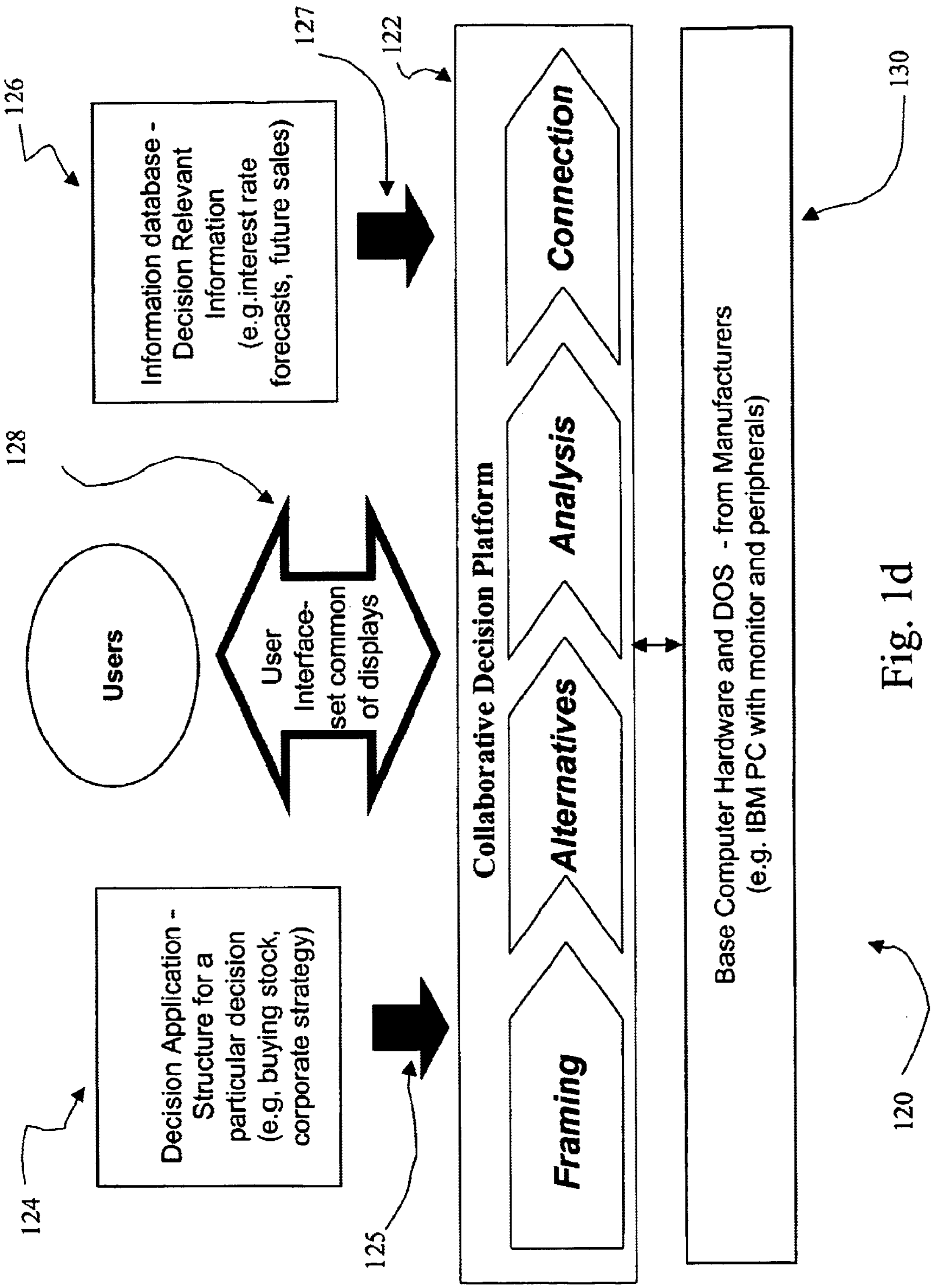

120
124
Decision Application - Structure for a particular decision (e.g, buying stock, corporate strategy)
125
Users
128
User Interface- set common of displays
126
Information database - Decision Relevant Information (e.g.interest rate forecasts, future sales)
127
122
Collaborative Decision Platform
Framing
Alternatives
Analysis
Connection
Base Computer Hardware and DOS - from Manufacturers (e.g. IBM PC with monitor and peripherals)
130

Fig. 1d

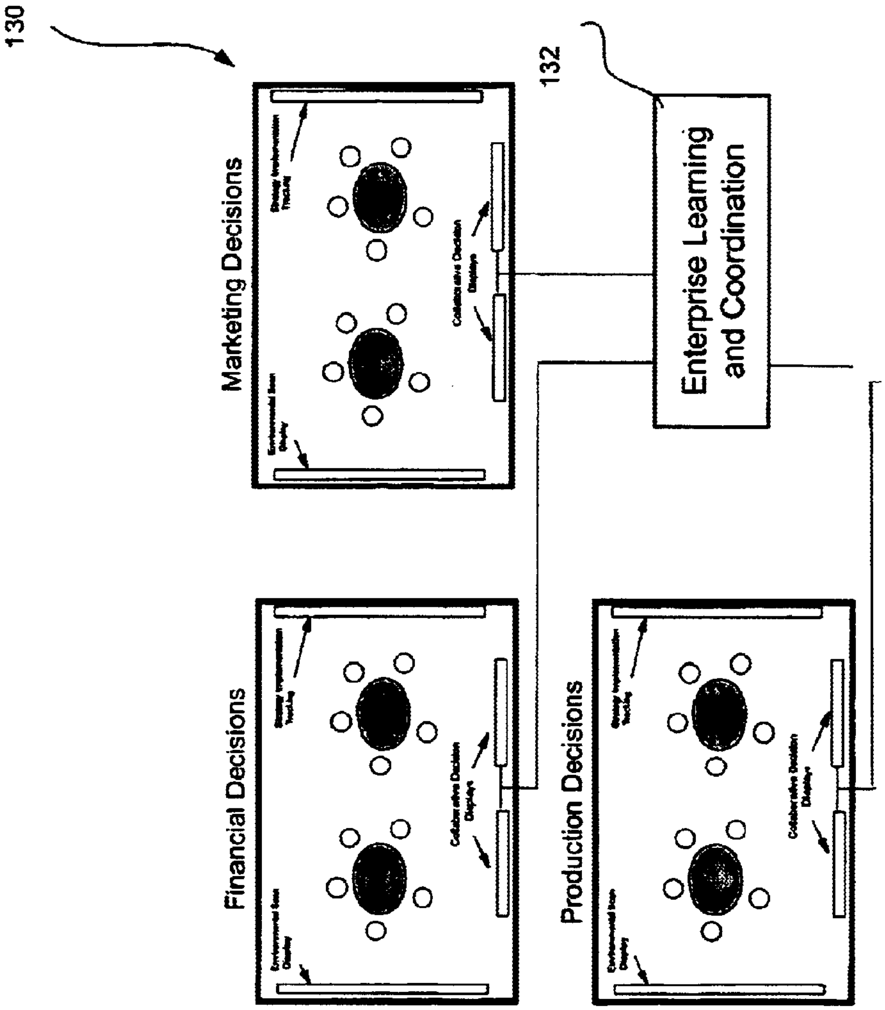

130
132
Marketing Decisions
Enterprise Learning and Coordination
Financial Decisions
Production Decisions

Fig. 1e

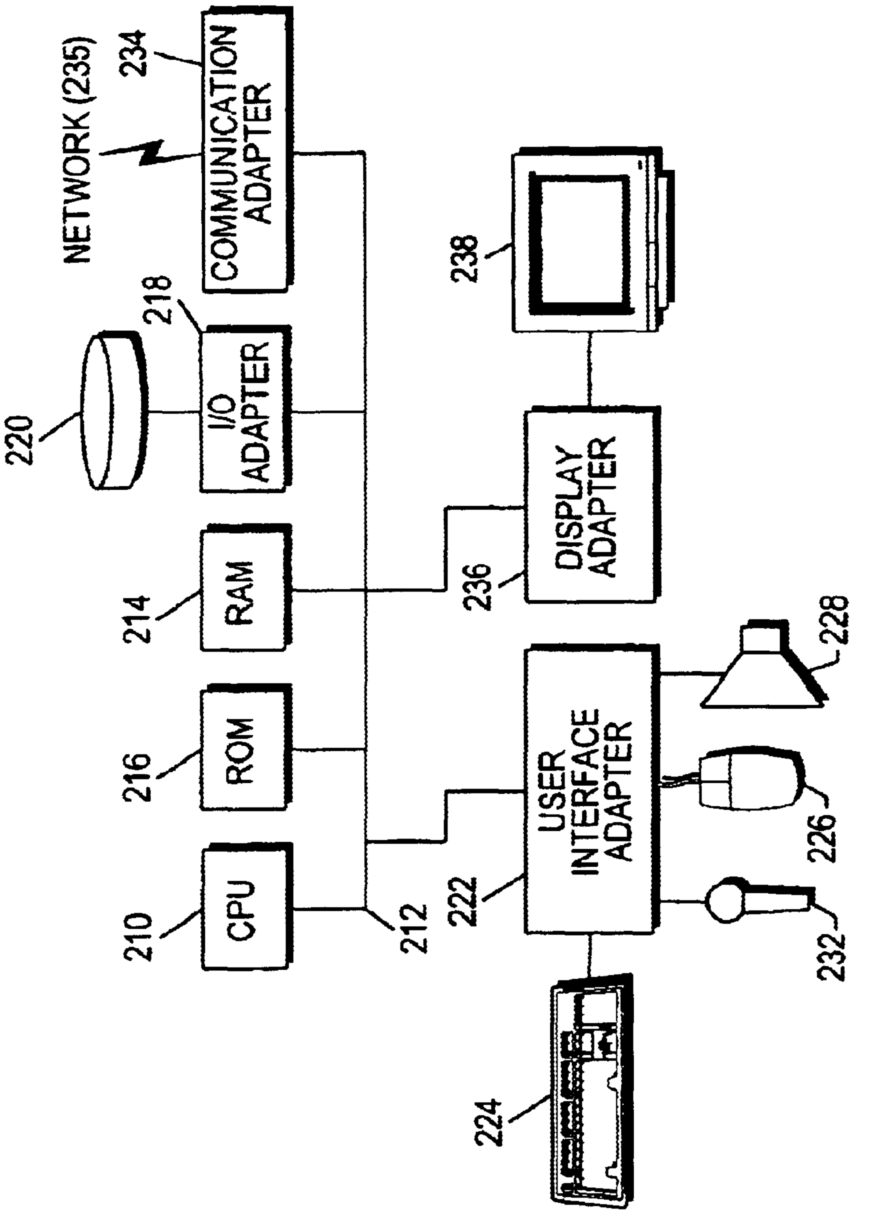

NETWORK (235)
234
COMMUNICATION ADAPTER
218
220
I/O ADAPTER
214
RAM
216
ROM
210
CPU
212
238
236
DISPLAY ADAPTER
222
USER INTERFACE ADAPTER
224
226
228
232

Fig. 2

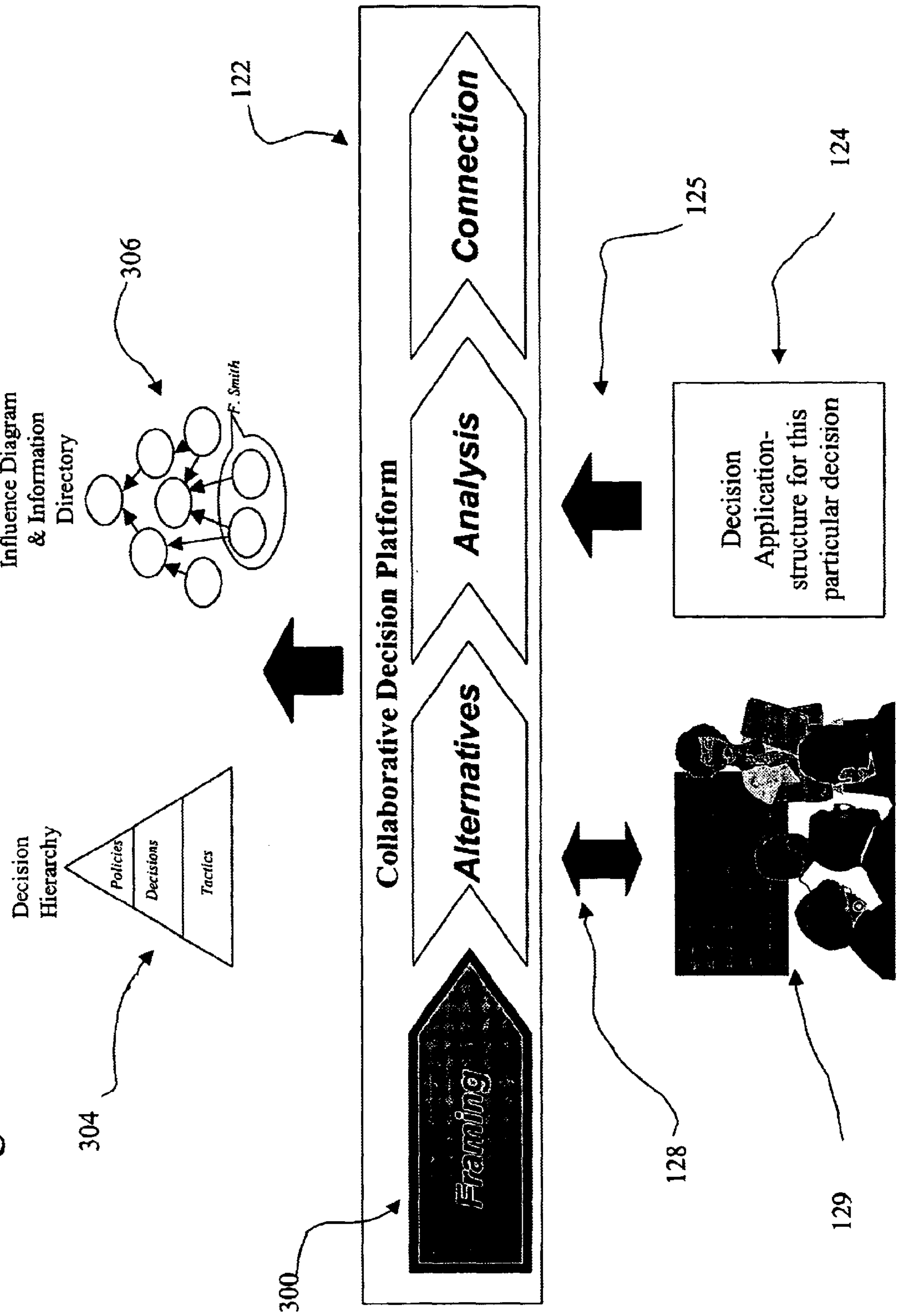

Decision Hierarchy
Policies
Decisions
Tactics
304
Influence Diagram & Information Directory
F. Smith
306
122
Collaborative Decision Platform
Framing
Alternatives
Analysis
Connection
300
125
124
Decision Application-structure for this particular decision
128
129

Fig. 3

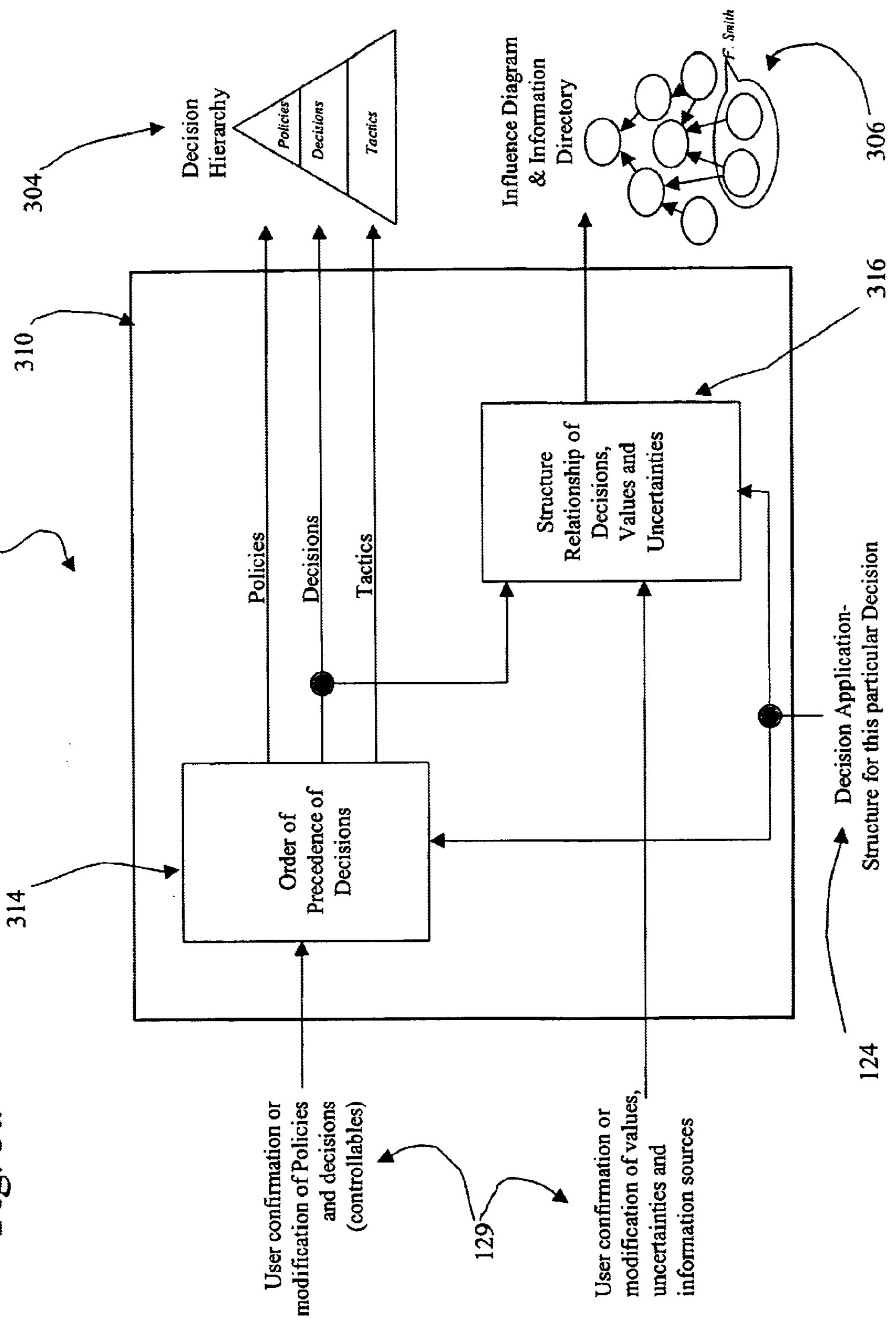

302
304
Decision Hierarchy
Policies
Decisions
Tactics
Influence Diagram & Information Directory
F. Smith
306
310
316
Policies
Decisions
Tactics
Structure Relationship of Decisions, Values and Uncertainties
314
Order of Precedence of Decisions
Decision Application-Structure for this particular Decision
124
User confirmation or modification of Policies and decisions (controllables)
129
User confirmation or modification of values, uncertainties and information sources

Fig. 3a

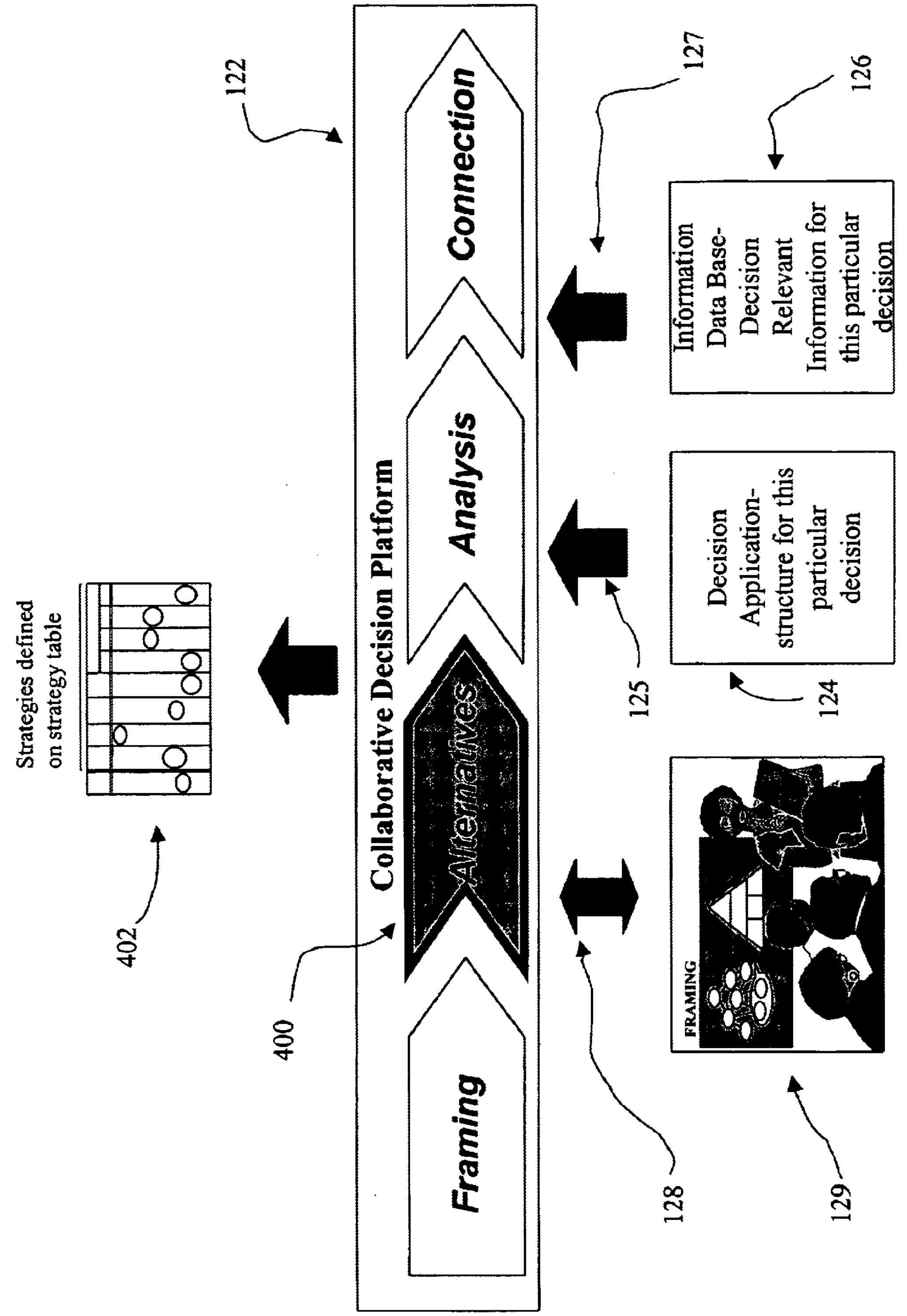

122
Collaborative Decision Platform
Framing
Alternatives
Analysis
Connection
400
402
Strategies defined on strategy table
127
126
Information Data Base-Decision Relevant Information for this particular decision
Decision Application-structure for this particular decision
125
124
FRAMING
128
129

Fig. 4

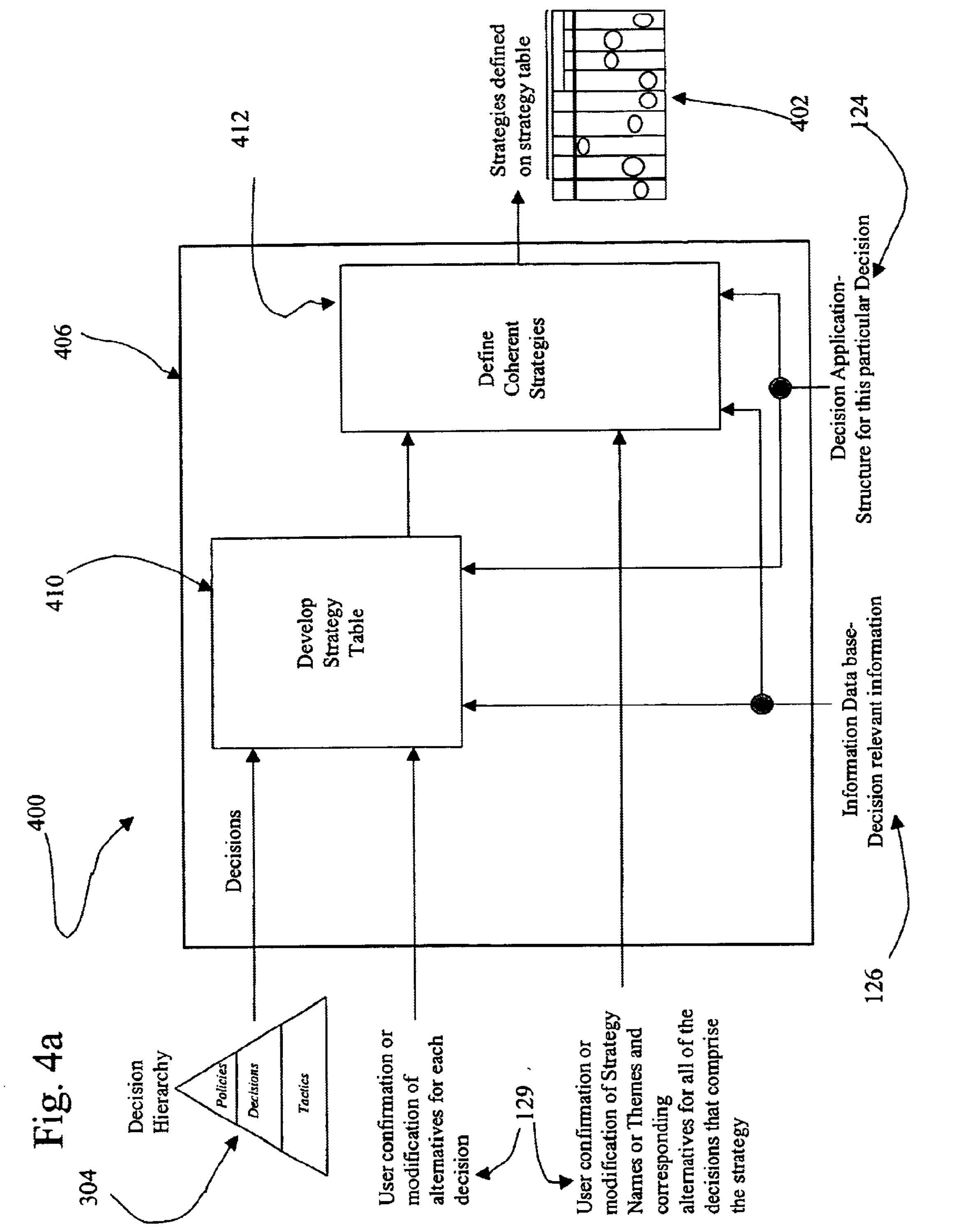

400
406
410
412
Decisions
Develop Strategy Table
Define Coherent Strategies
Strategies defined on strategy table
402
124
Decision Application-Structure for this particular Decision
Information Data base-Decision relevant information
126
Decision Hierarchy
Policies
Decisions
Tactics
304
User confirmation or modification of alternatives for each decision
129
User confirmation or modification of Strategy Names or Themes and corresponding alternatives for all of the decisions that comprise the strategy

Fig. 4a

Fig. 5

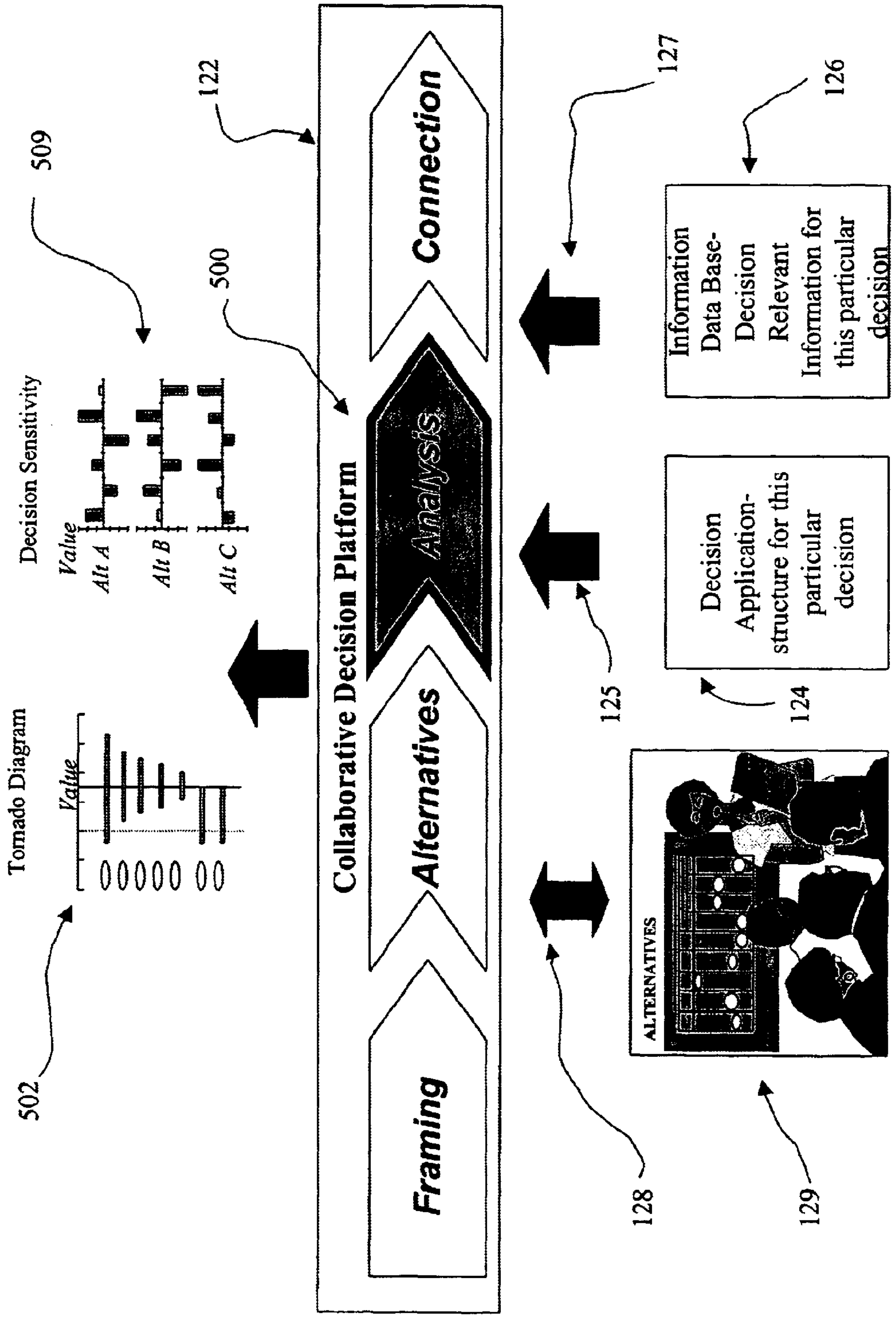
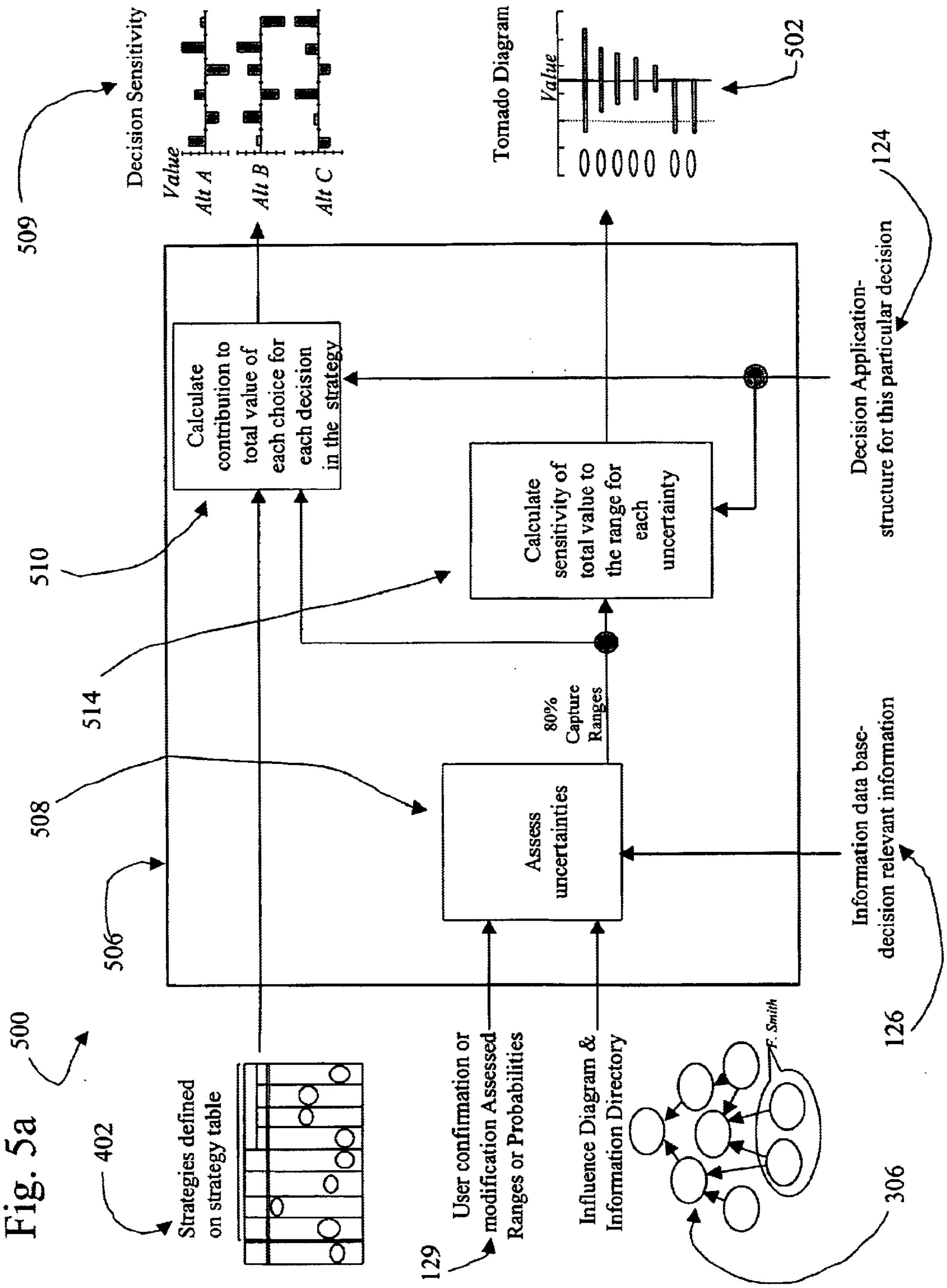
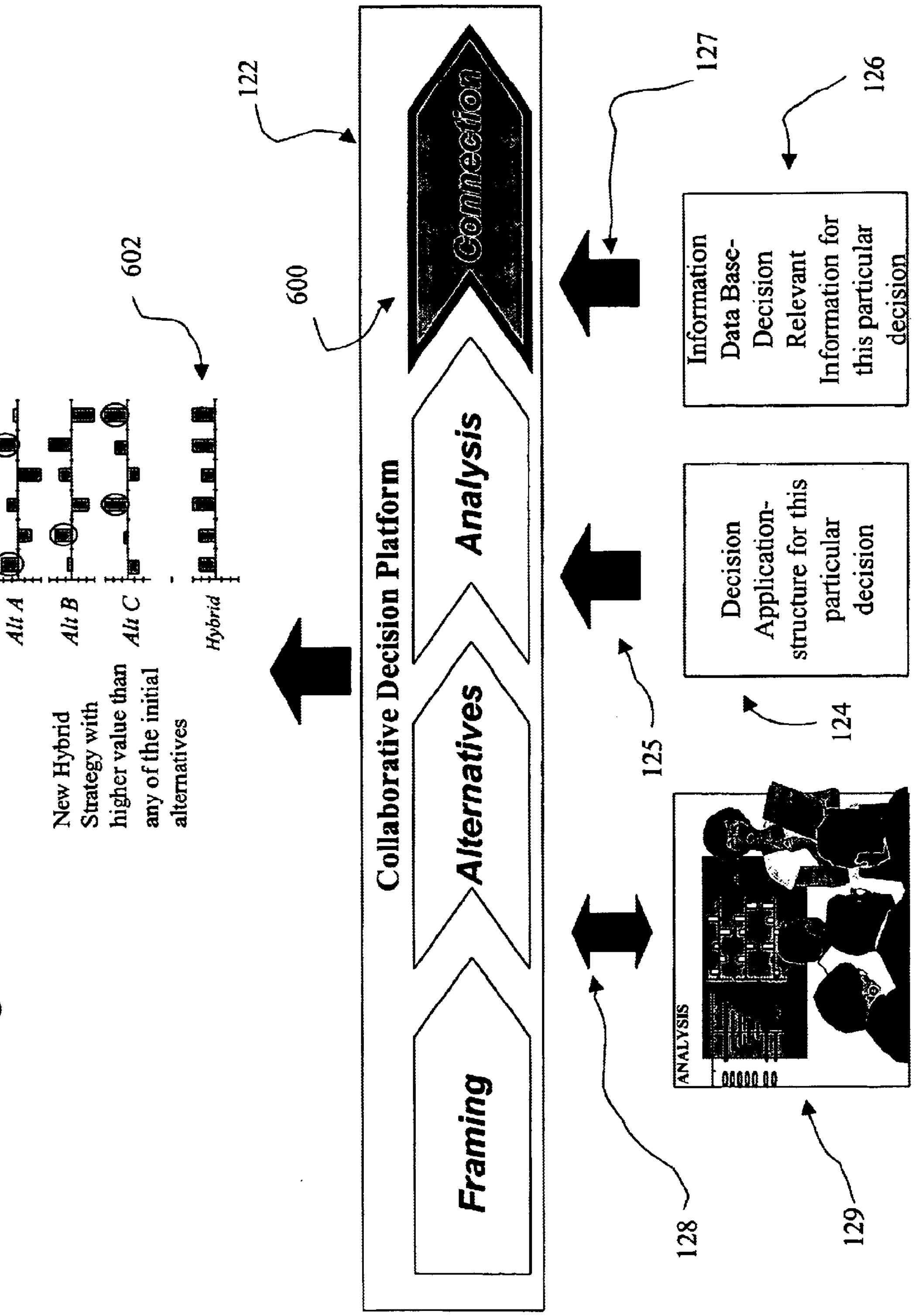

502
Tornado Diagram
Value
509
Decision Sensitivity
Value
Alt A
Alt B
Alt C
122
500
Collaborative Decision Platform
Framing
Alternatives
Analysis
Connection
127
126
125
124
128
129
ALTERNATIVES
Decision Application-structure for this particular decision
Information Data Base-Decision Relevant Information for this particular decision

Fig. 5a


500
402
Strategies defined on strategy table
129
User confirmation or modification Assessed Ranges or Probabilities
Influence Diagram & Information Directory
F. Smith
306
126
Information data base- decision relevant information
506
508
Assess uncertainties
80% Capture Ranges
514
Calculate sensitivity of total value to the range for each uncertainty
510
Calculate contribution to total value of each choice for each decision in the strategy
Decision Application- structure for this particular decision
124
509
Decision Sensitivity
Value
Alt A
Alt B
Alt C
Tornado Diagram
Value
502



Collaborative Decision Platform
Framing
Alternatives
Analysis
Connection
122
600
602
127
126
125
124
128
129
Information Data Base-Decision Relevant Information for this particular decision
Decision Application-structure for this particular decision
ANALYSIS
Value
Alt A
Alt B
Alt C
Hybrid
New Hybrid Strategy with higher value than any of the initial alternatives

Fig. 6

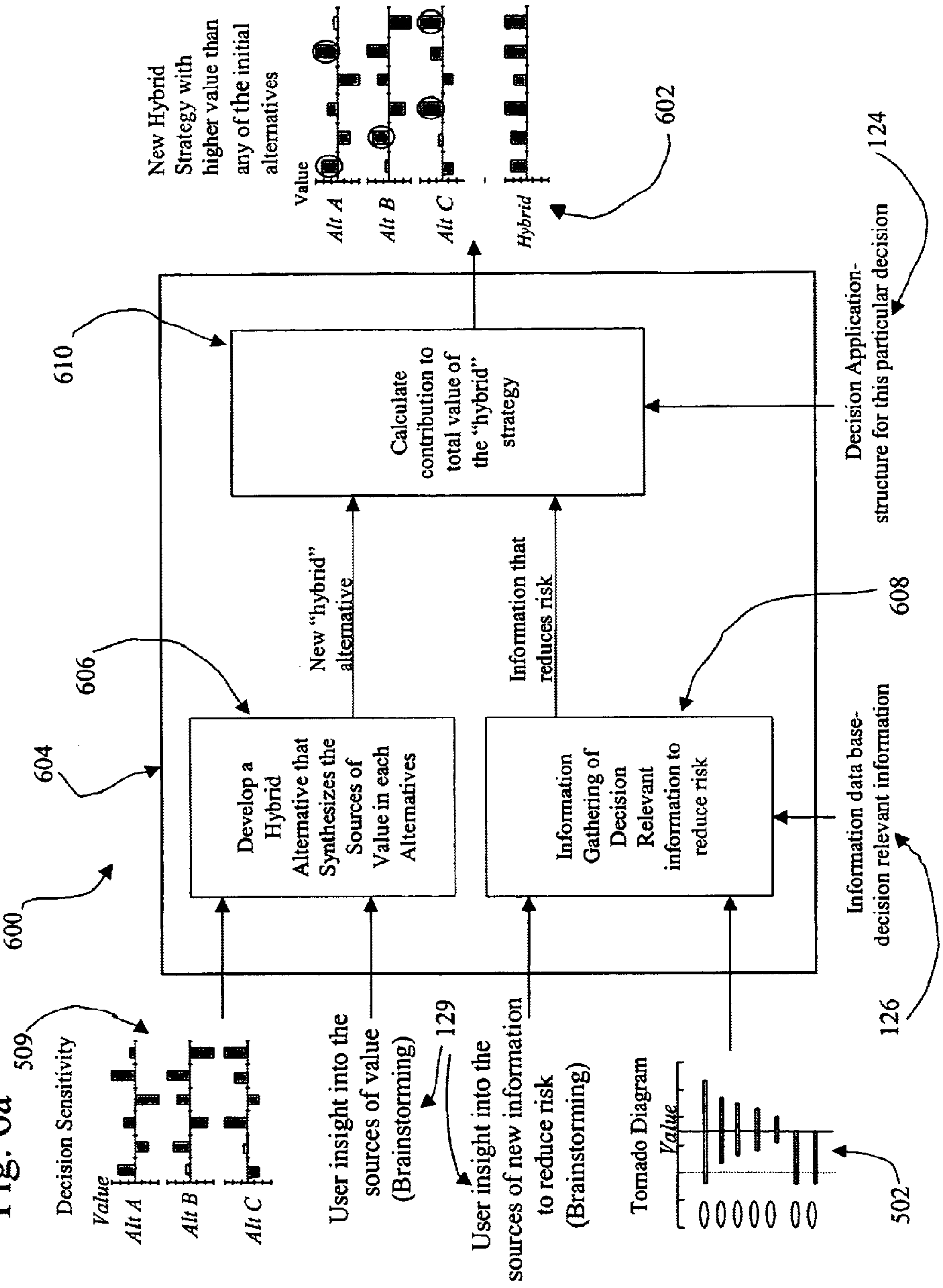

509
Decision Sensitivity
Value
Alt A
Alt B
Alt C
User insight into the sources of value (Brainstorming)
129
User insight into the sources of new information to reduce risk (Brainstorming)
Tornado Diagram
Value
502
600
604
606
Develop a Hybrid Alternative that Synthesizes the Sources of Value in each Alternatives
Information Gathering of Decision Relevant information to reduce risk
Information data base-decision relevant information
126
New "hybrid" alternative
Information that reduces risk
608
610
Calculate contribution to total value of the "hybrid" strategy
Decision Application-structure for this particular decision
124
New Hybrid Strategy with higher value than any of the initial alternatives
Value
Alt A
Alt B
Alt C
Hybrid
602

Fig. 6a

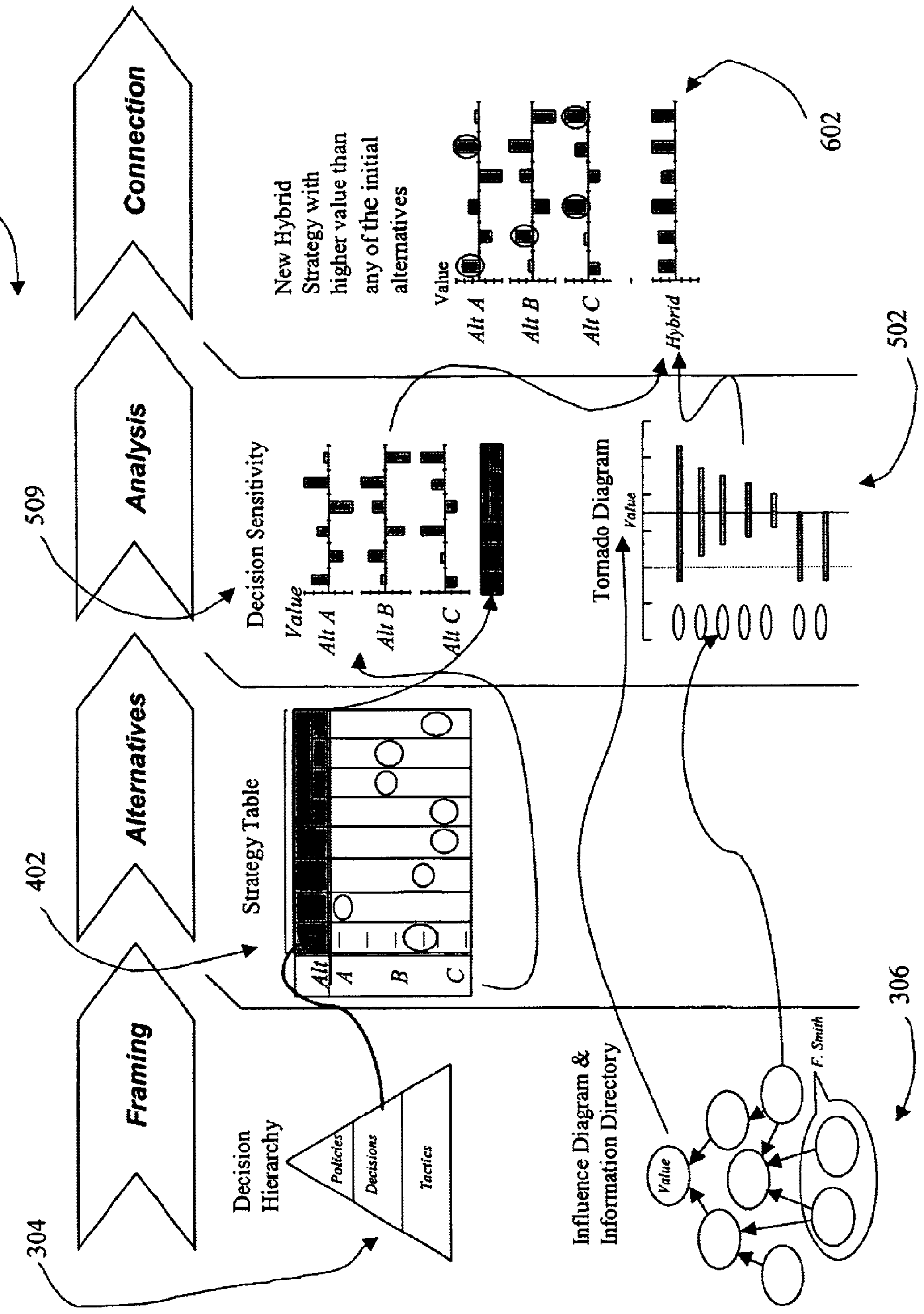
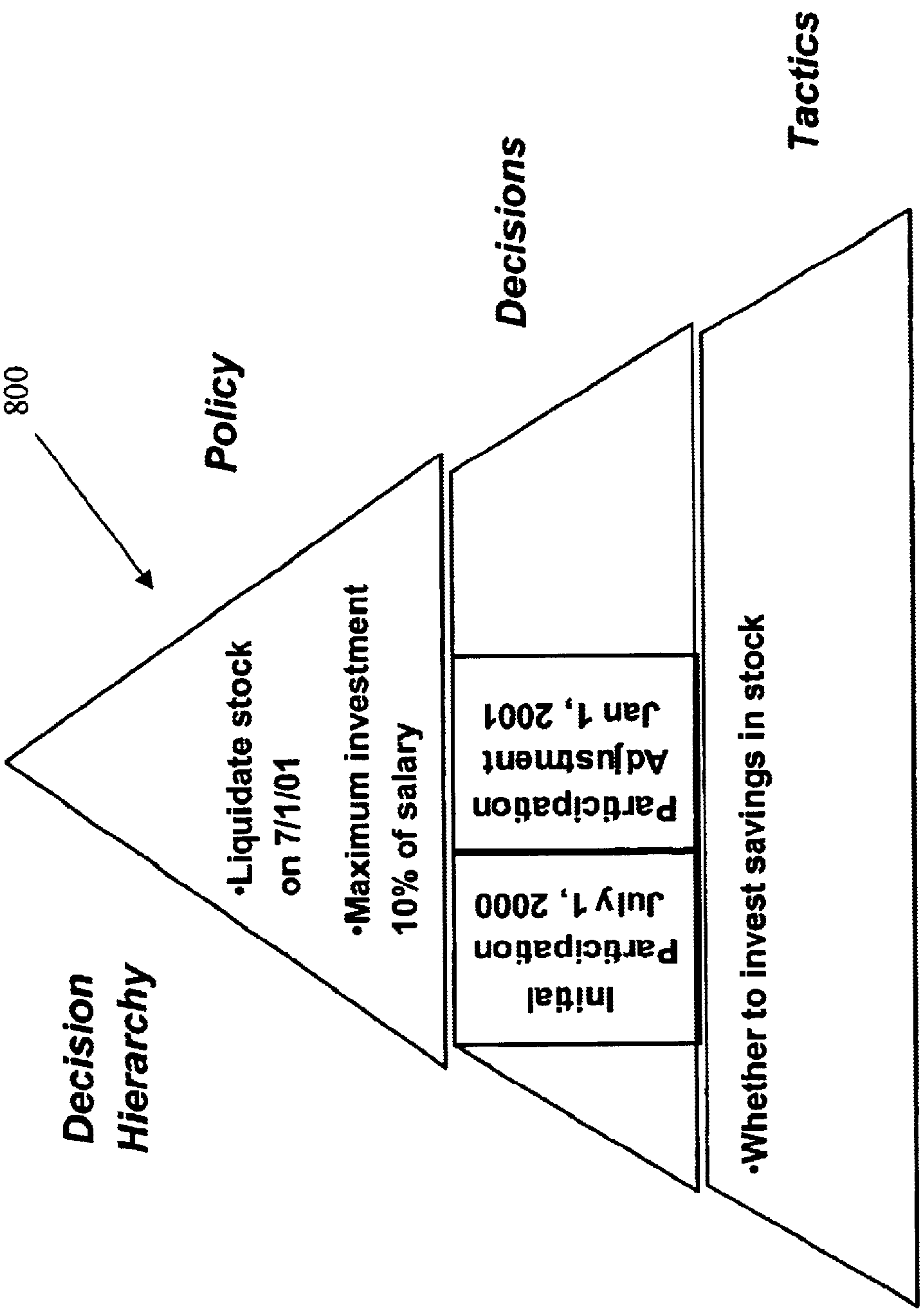

128
Framing
Alternatives
Analysis
Connection
304
402
509
Decision Hierarchy
Policies
Decisions
Tactics
Strategy Table
Alt
A
B
C
Decision Sensitivity
Value
Alt A
Alt B
Alt C
New Hybrid Strategy with higher value than any of the initial alternatives
Value
Alt A
Alt B
Alt C
Hybrid
602
Tornado Diagram
Value
502
Influence Diagram & Information Directory
Value
F. Smith
306

Fig. 7



800
Decision Hierarchy
Policy
Decisions
Tactics
•Liquidate stock on 7/1/01
•Maximum investment 10% of salary
Initial Participation July 1, 2000
Participation Adjustment Jan 1, 2001
•Whether to invest savings in stock

Fig. 8a

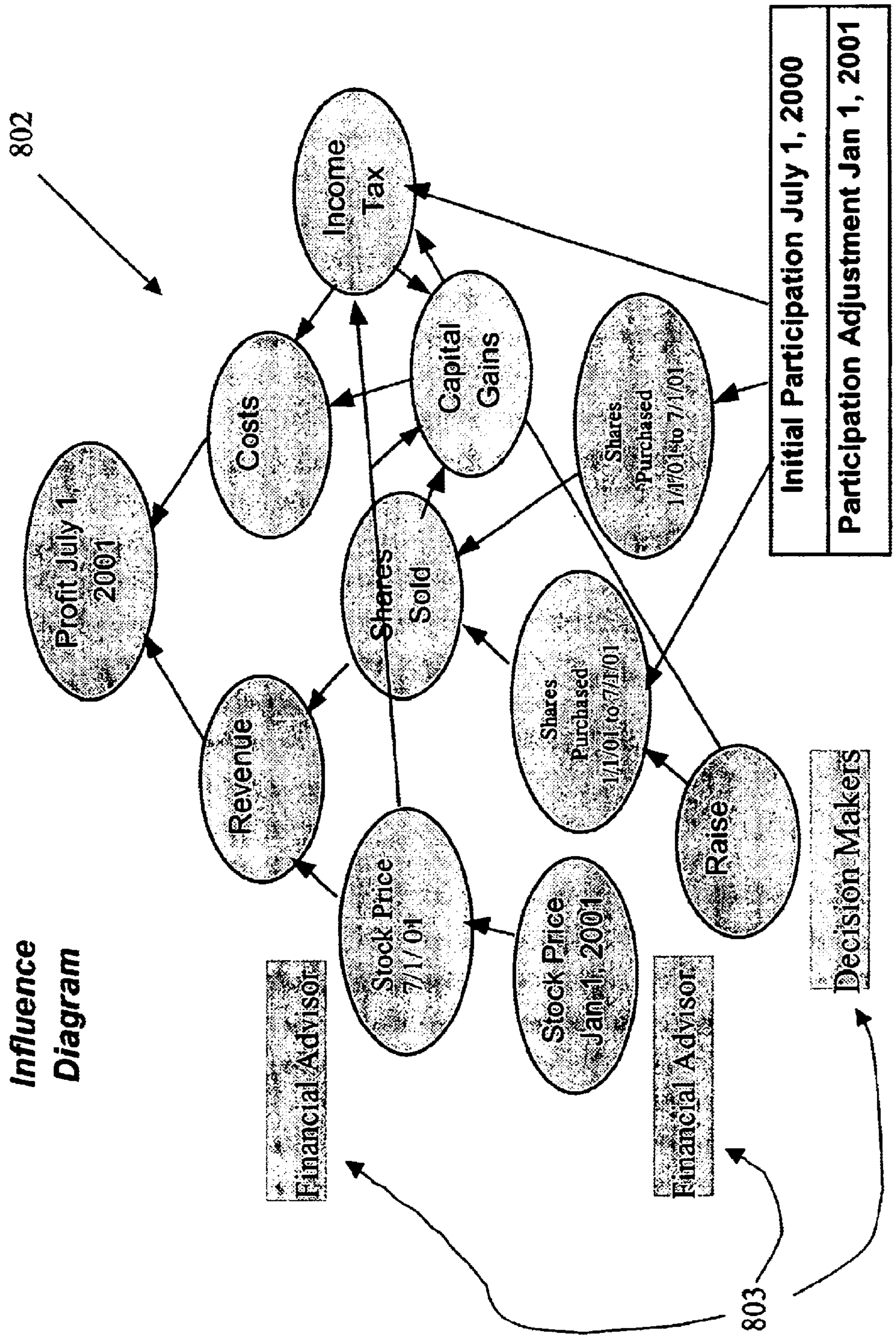

Influence Diagram
802
Income Tax
Costs
Capital Gains
Profit July 1, 2001
Shares Sold
Shares Purchased 1/1/01 to 7/1/01
Shares Purchased 1/1/01 to 7/1/01
Revenue
Stock Price 7/1/01
Stock Price Jan 1, 2001
Raise
Financial Advisor
Financial Advisor
Decision Makers
Initial Participation July 1, 2000
Participation Adjustment Jan 1, 2001
803

Fig. 8b

804

Strategy Table

Strategy Name	Initial Participation July 1, 2000	Participation Adjustment Jan 1, 2001
No Participation	0%	0%
In and Out	5%	5%
Out and In	7%	7%
	10%	10%

Fig. 8c

804

Strategy Table

Strategy Name	Initial Participation July 1, 2000	Participation Adjustment Jan 1, 2001
No Participation	0%	0%
In and Out	5%	5%
Out and In	7%	7%
	10%	10%

Fig. 8d

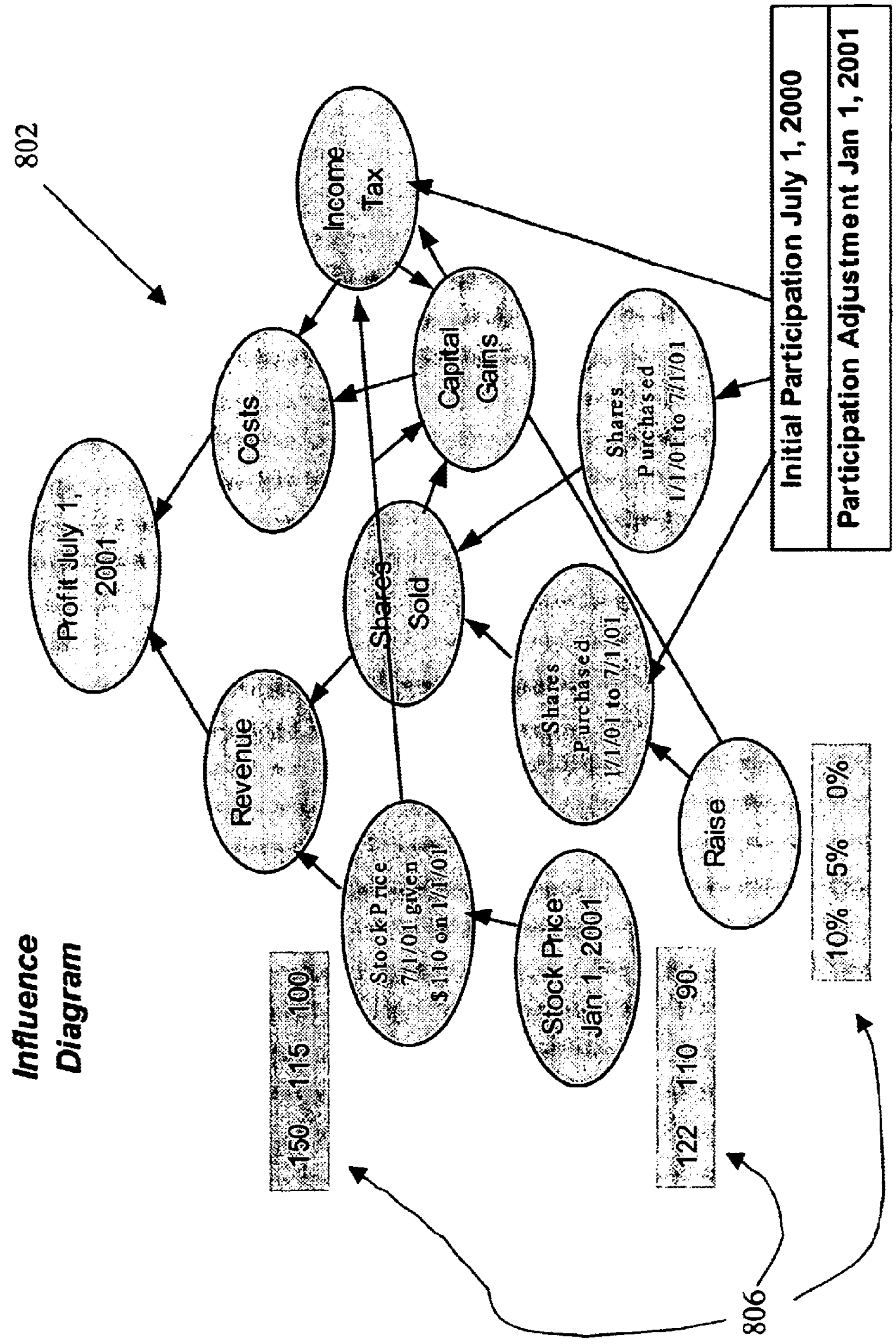

802
Influence Diagram
Income Tax
Costs
Capital Gains
Profit July 1, 2001
Shares Sold
Shares Purchased 1/1/01 to 7/1/01
Shares Purchased 1/1/01 to 7/1/01
Revenue
Stock Price 7/1/01 given $110 on 1/1/01
Stock Price Jan 1, 2001
Raise
150 115 100
122 110 90
10% 5% 0%
Initial Participation July 1, 2000
Participation Adjustment Jan 1, 2001
806

Fig. 8e

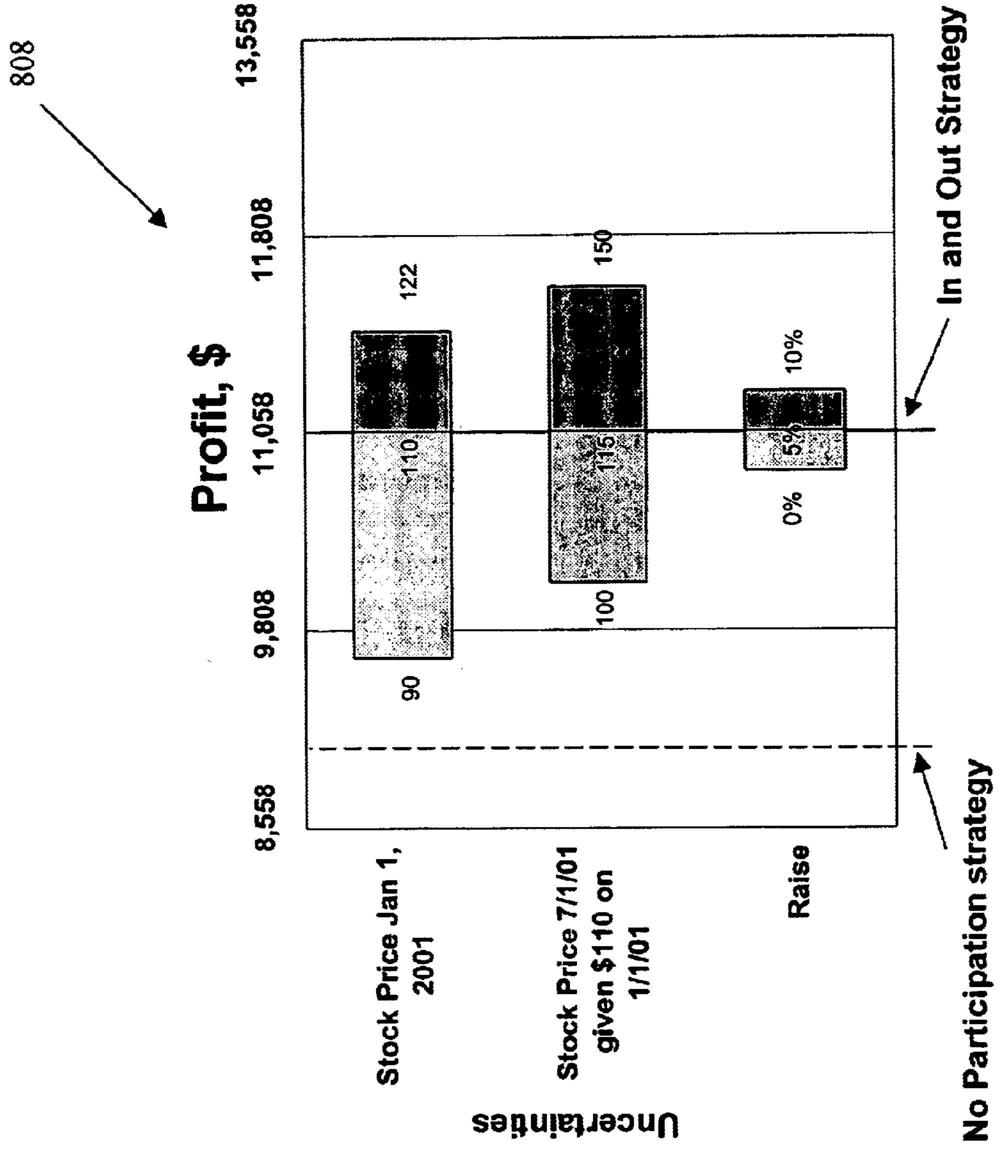

808
Profit, $
8,558
9,808
11,058
11,808
13,558
90
110
122
100
115
150
0%
5%
10%
Stock Price Jan 1, 2001
Stock Price 7/1/01 given $110 on 1/1/01
Raise
Uncertainties
In and Out Strategy
No Participation strategy

Fig. 8f

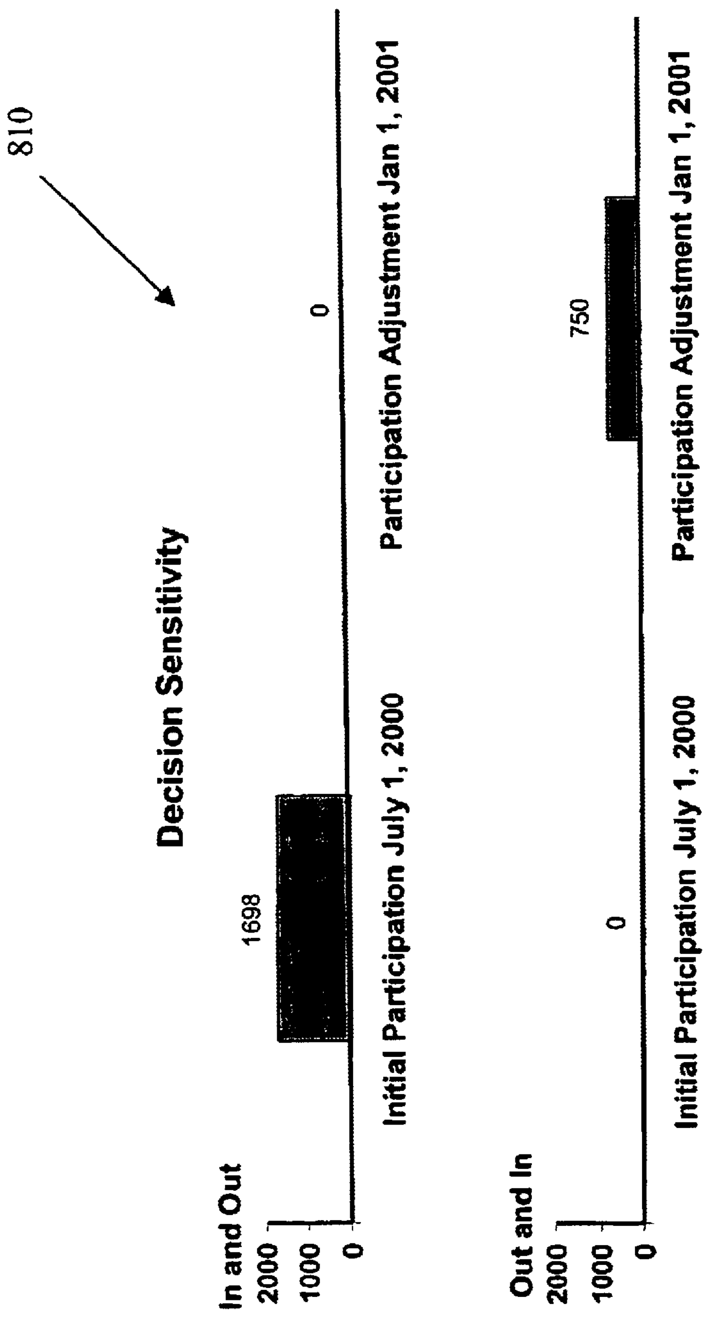

810
Decision Sensitivity
In and Out
2000
1000
0
1698
0
Initial Participation July 1, 2000
Participation Adjustment Jan 1, 2001
Out and In
2000
1000
0
0
750
Initial Participation July 1, 2000
Participation Adjustment Jan 1, 2001

Fig. 8g

804

Strategy Table

Strategy Name	Initial Participation July 1, 2000	Participation Adjustment Jan 1, 2001
No Participation	0%	0%
In and Out	5%	5%
Out and In	7%	7%
Maximum Contribution	10%	10%

811

Fig. 8h

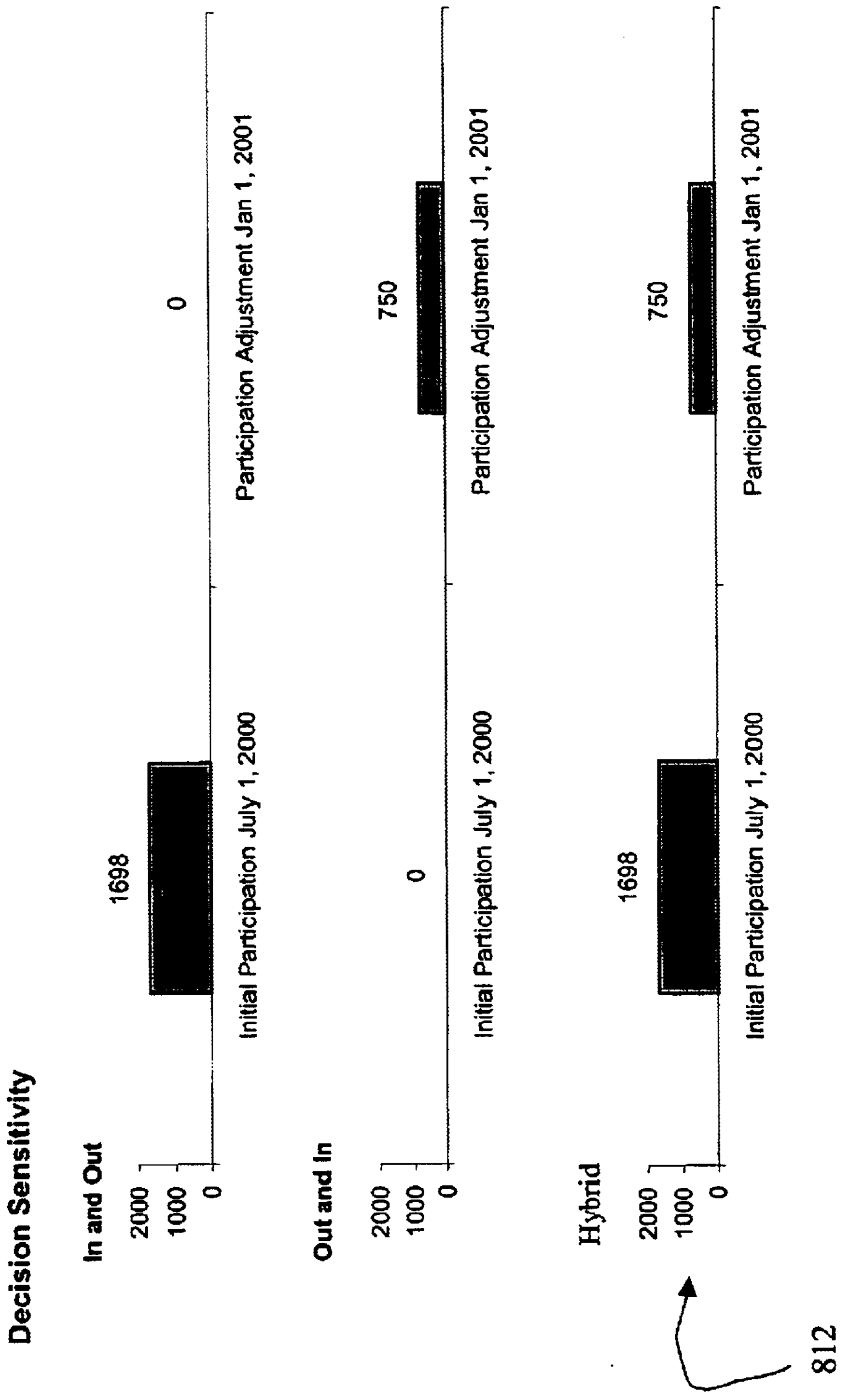

Decision Sensitivity
In and Out
2000
1000
0
1698
Initial Participation July 1, 2000
0
Participation Adjustment Jan 1, 2001
Out and In
2000
1000
0
0
Initial Participation July 1, 2000
750
Participation Adjustment Jan 1, 2001
Hybrid
2000
1000
0
1698
Initial Participation July 1, 2000
750
Participation Adjustment Jan 1, 2001
812

Fig. 8i

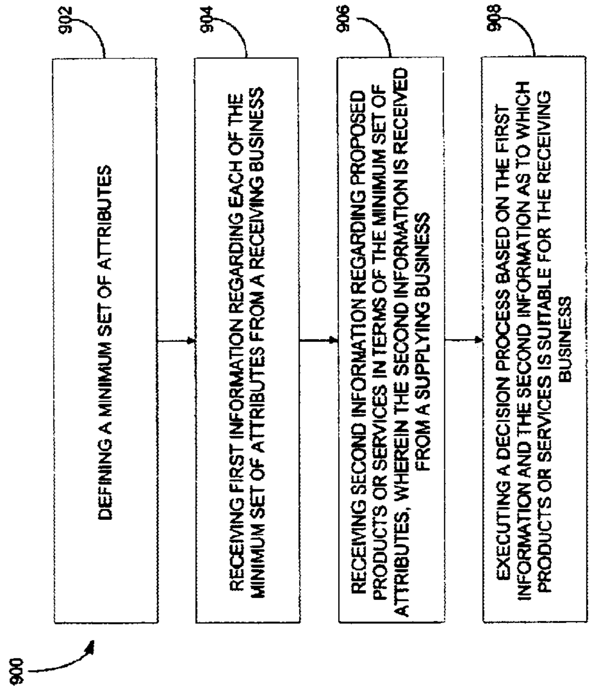

900
902
DEFINING A MINIMUM SET OF ATTRIBUTES
904
RECEIVING FIRST INFORMATION REGARDING EACH OF THE MINIMUM SET OF ATTRIBUTES FROM A RECEIVING BUSINESS
906
RECEIVING SECOND INFORMATION REGARDING PROPOSED PRODUCTS OR SERVICES IN TERMS OF THE MINIMUM SET OF ATTRIBUTES, WHEREIN THE SECOND INFORMATION IS RECEIVED FROM A SUPPLYING BUSINESS
908
EXECUTING A DECISION PROCESS BASED ON THE FIRST INFORMATION AND THE SECOND INFORMATION AS TO WHICH PRODUCTS OR SERVICES IS SUITABLE FOR THE RECEIVING BUSINESS

Fig. 9

920

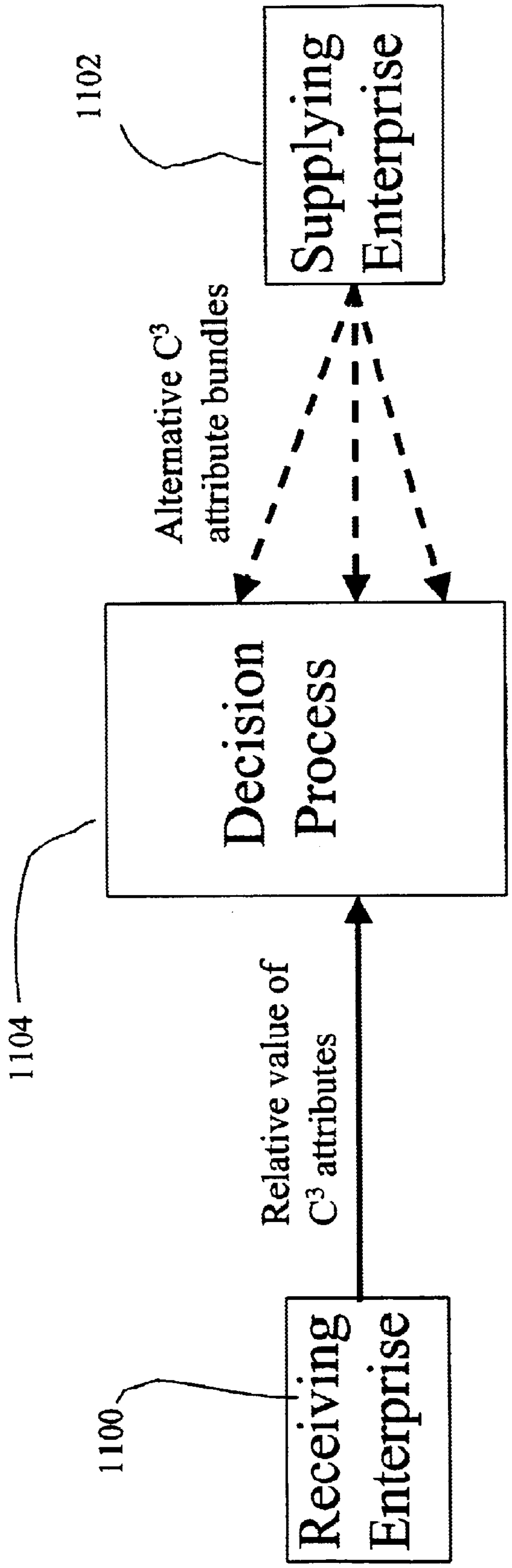
	Value of one percent increase in C^3 Attribute	
C^3 Attribute	Typical Automotive Manufacturer	Typical Information Technology Company
Revenue ($M)	200	15
COGS ($M)	75	35
SG&A ($M)	20	40
Inventory Turns	12	15
Receivable Turns	2	2

Fig. 9a

1000

C^3 Attribute	Percent increase in C^3 Attribute	
	Alternative 1: Improve Supply Chain Management	Alternative 2: Installation of Customer Relationship Management system
Revenue	0	5
COGS	-4	0
SG&A	-1	0
Inventory Turns	+5	0
Receivable Turns	+1	+2

Fig. 10



1102
Supplying Enterprise
Alternative C^3 attribute bundles
1104
Decision Process
Relative value of C^3 attributes
1100
Receiving Enterprise

Fig. 11

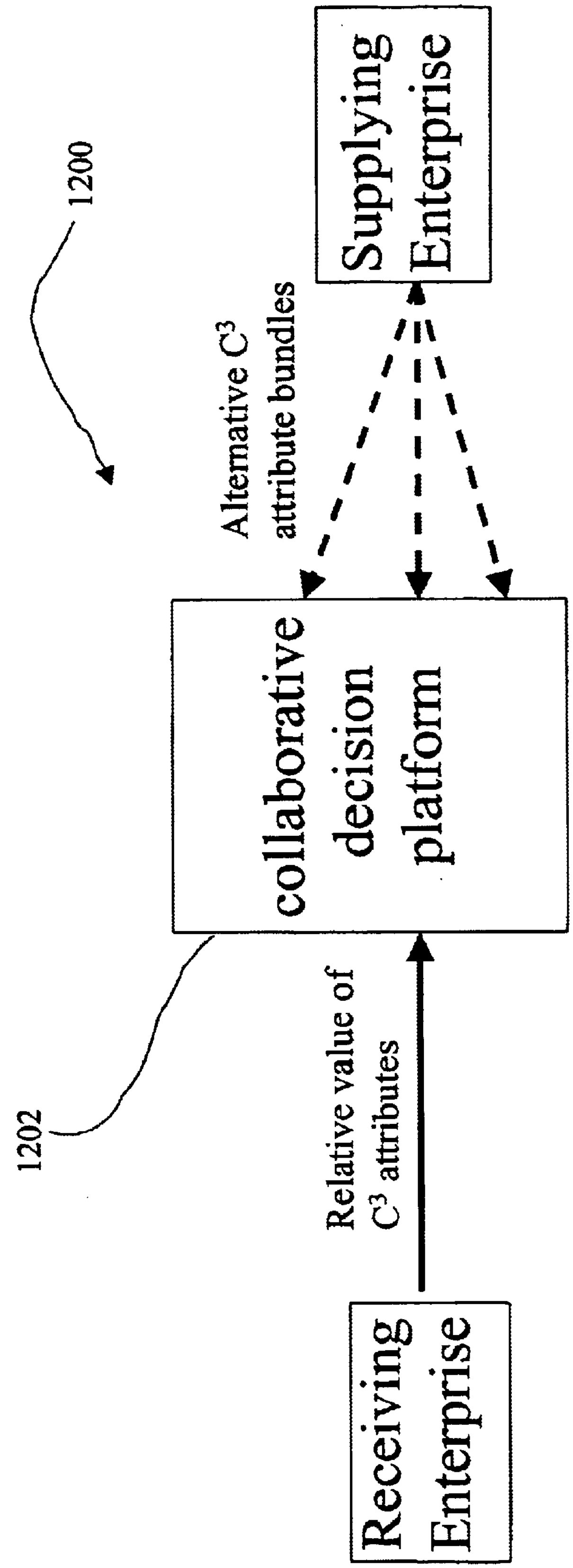

1200
Supplying Enterprise
Alternative C^3 attribute bundles
collaborative decision platform
1202
Relative value of C^3 attributes
Receiving Enterprise

Fig. 12

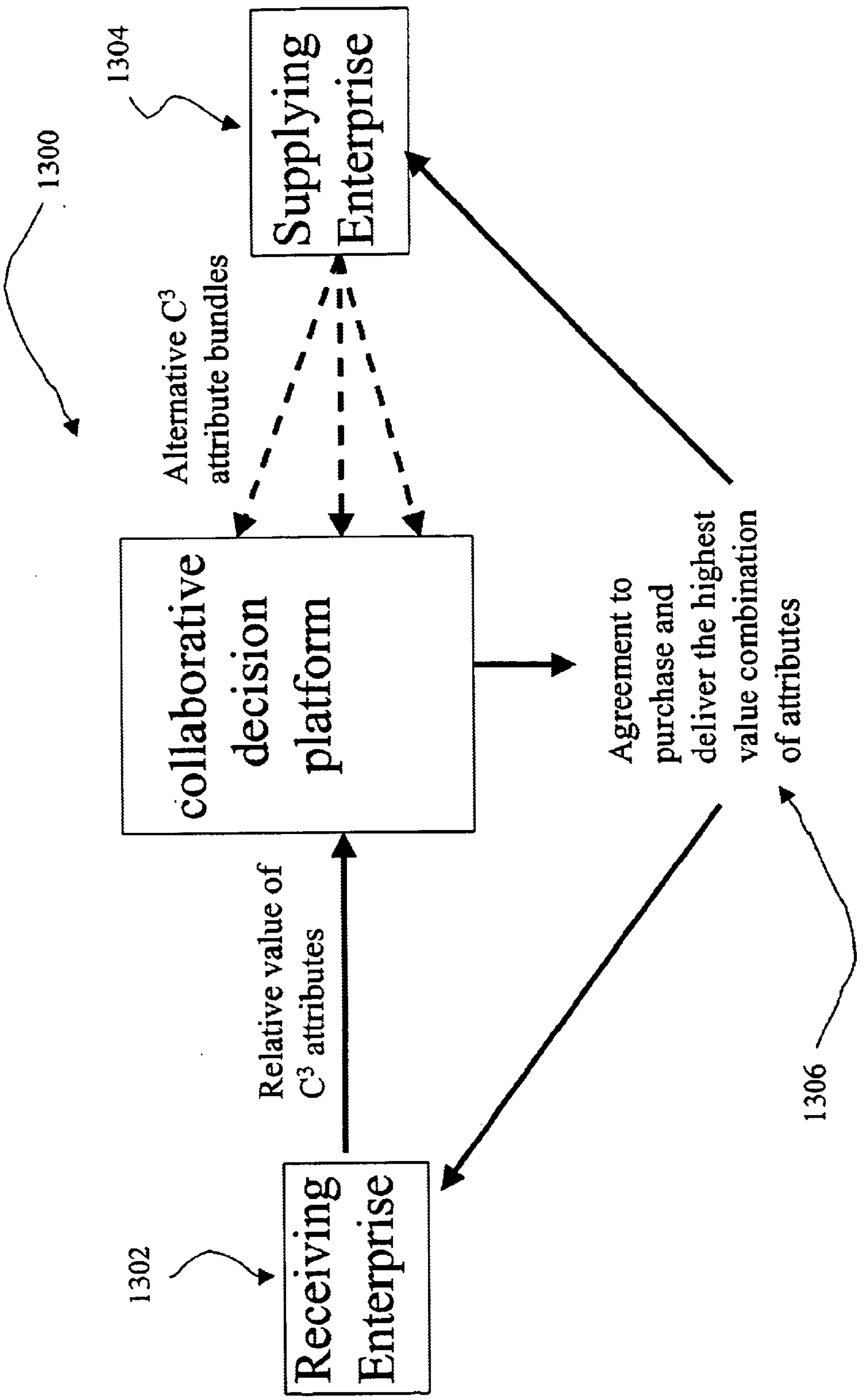

1300
1304
Supplying Enterprise
Alternative C3 attribute bundles
collaborative decision platform
Agreement to purchase and deliver the highest value combination of attributes
1306
Relative value of C3 attributes
1302
Receiving Enterprise

Fig. 13

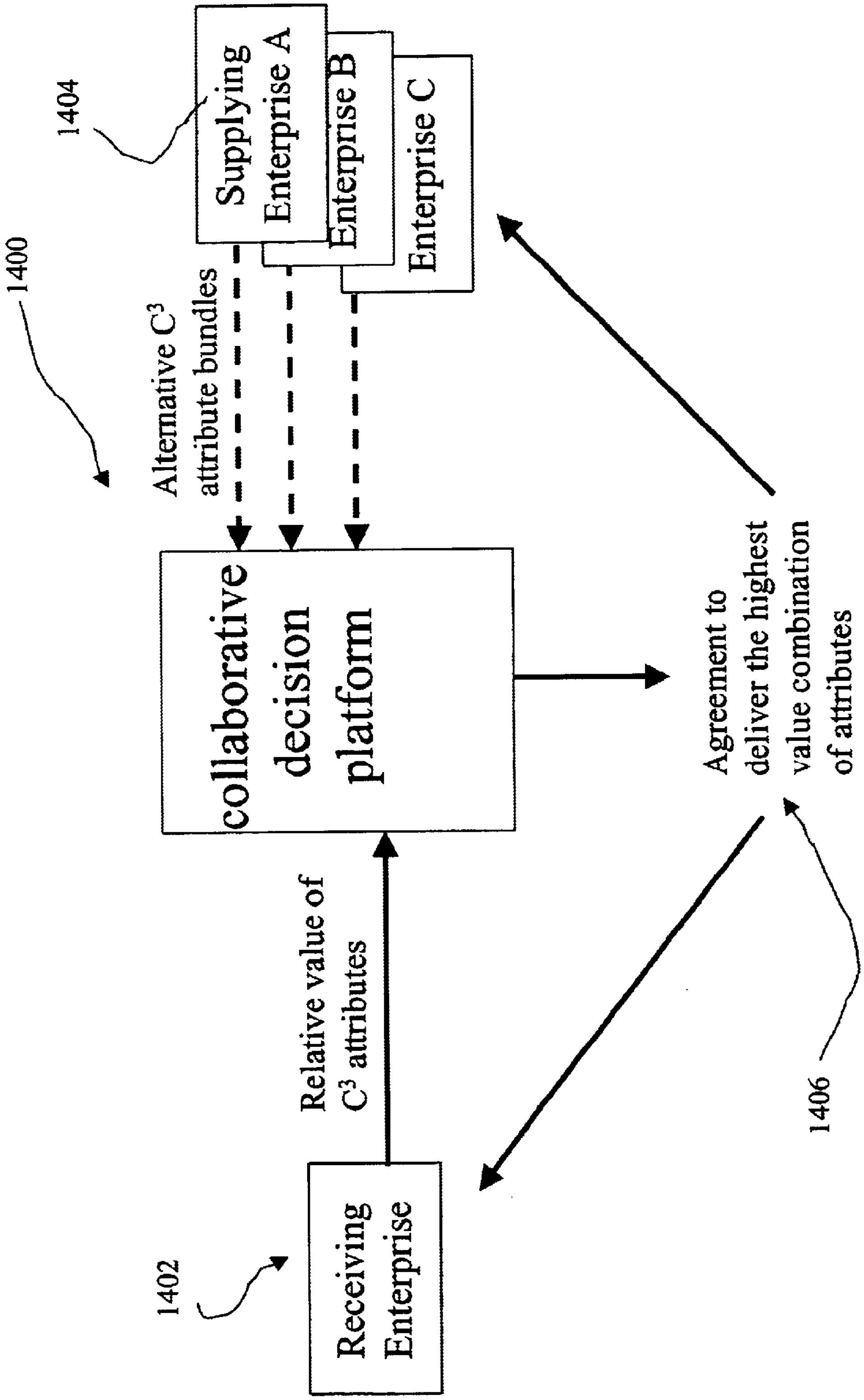

1400
1404
Supplying Enterprise A
Enterprise B
Enterprise C
Alternative C3 attribute bundles
collaborative decision platform
Agreement to deliver the highest value combination of attributes
Relative value of C3 attributes
Receiving Enterprise
1402
1406

Fig. 14

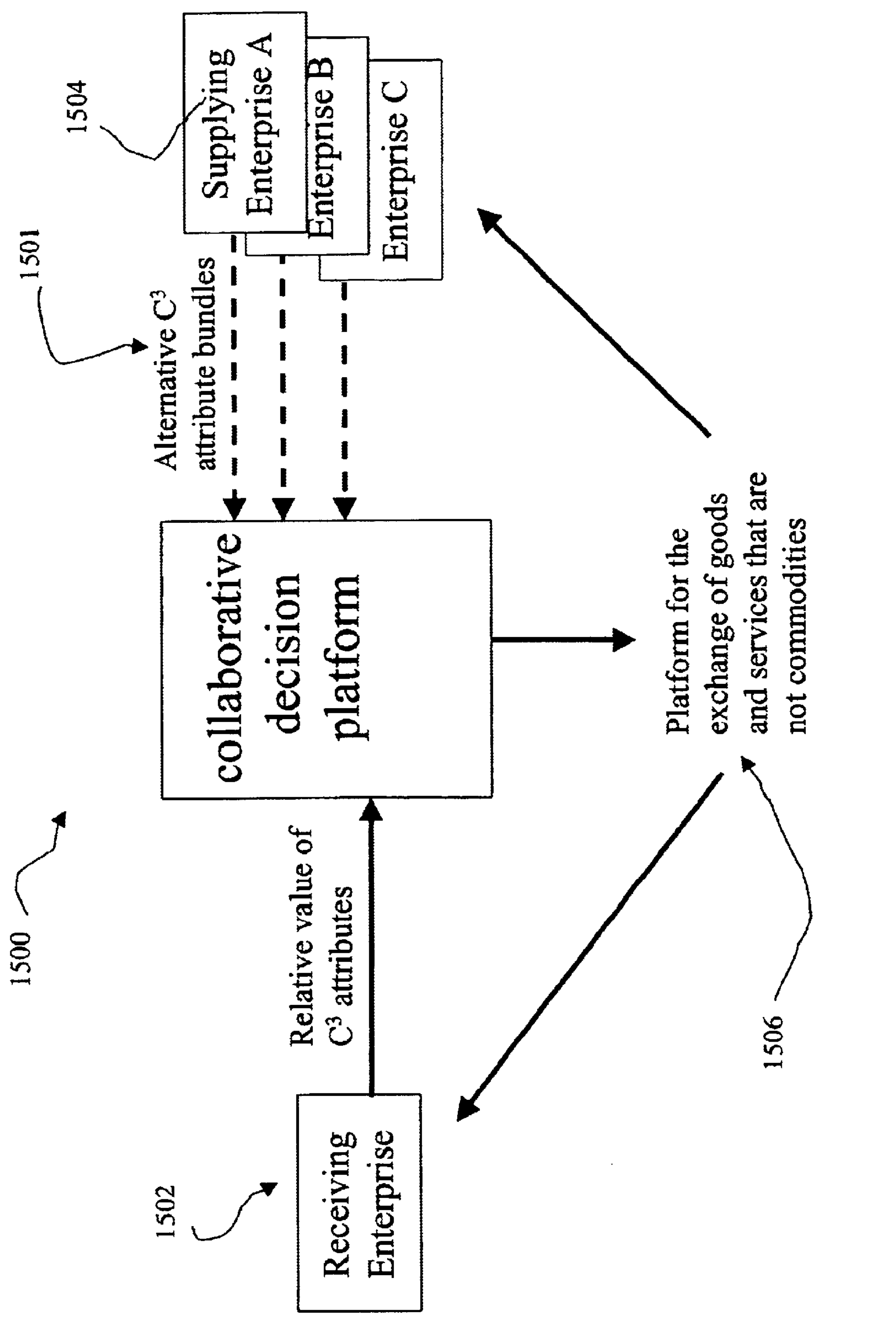

1500
1501
1502
1504
1506
Supplying Enterprise A
Enterprise B
Enterprise C
Alternative C3 attribute bundles
collaborative decision platform
Relative value of C3 attributes
Receiving Enterprise
Platform for the exchange of goods and services that are not commodities

Fig. 15

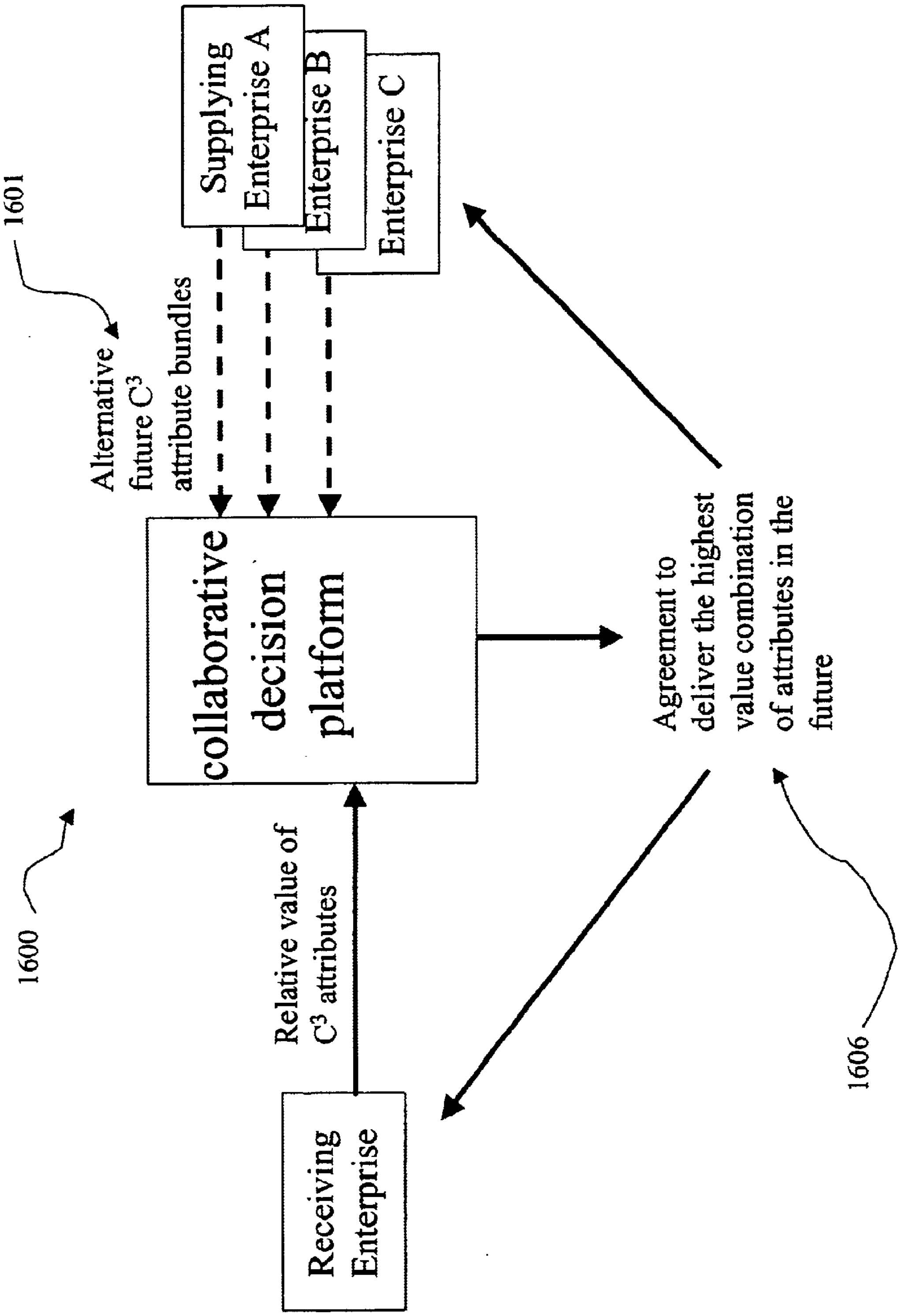

1601
Supplying Enterprise A
Enterprise B
Enterprise C
Alternative future C3 attribute bundles
collaborative decision platform
Agreement to deliver the highest value combination of attributes in the future
1600
Relative value of C3 attributes
Receiving Enterprise
1606

Fig. 16

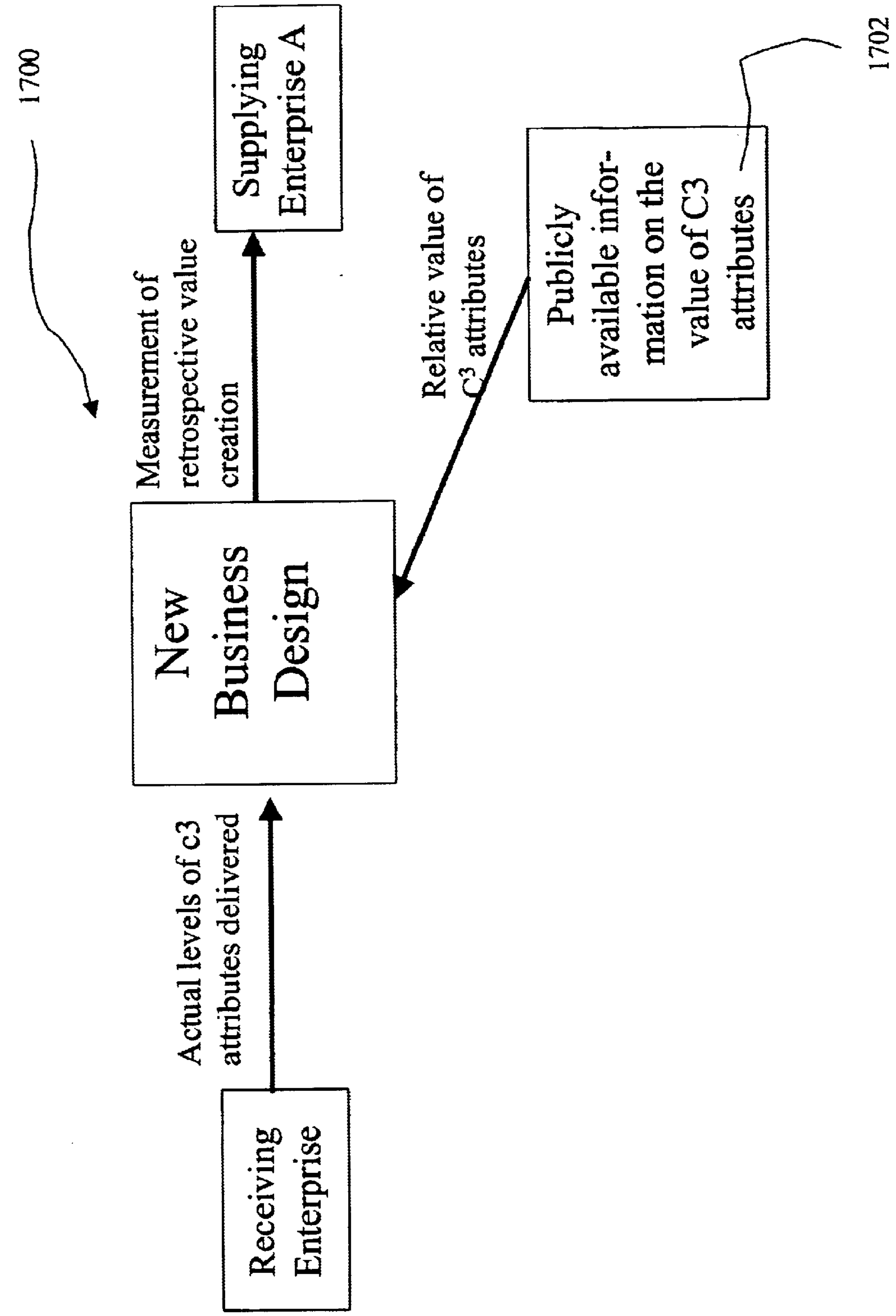

1700
Supplying Enterprise A
Measurement of retrospective value creation
Relative value of C3 attributes
Publicly available infor-mation on the value of C3 attributes
1702
New Business Design
Actual levels of c3 attributes delivered
Receiving Enterprise

Fig. 17

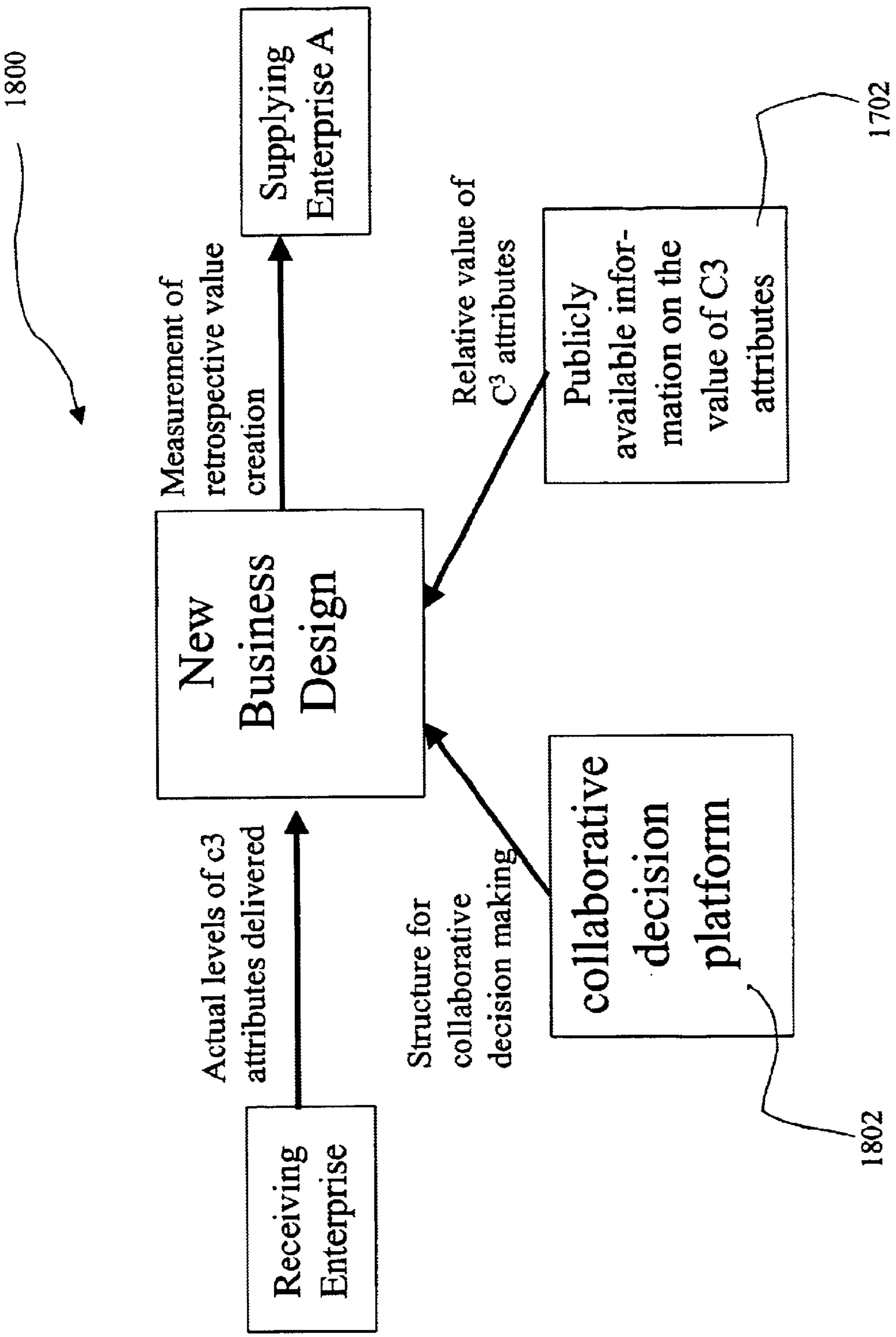

1800
Supplying Enterprise A
Measurement of retrospective value creation
New Business Design
Relative value of C^3 attributes
Publicly available information on the value of C3 attributes
1702
Actual levels of c3 attributes delivered
Receiving Enterprise
Structure for collaborative decision making
collaborative decision platform
1802

Fig. 18

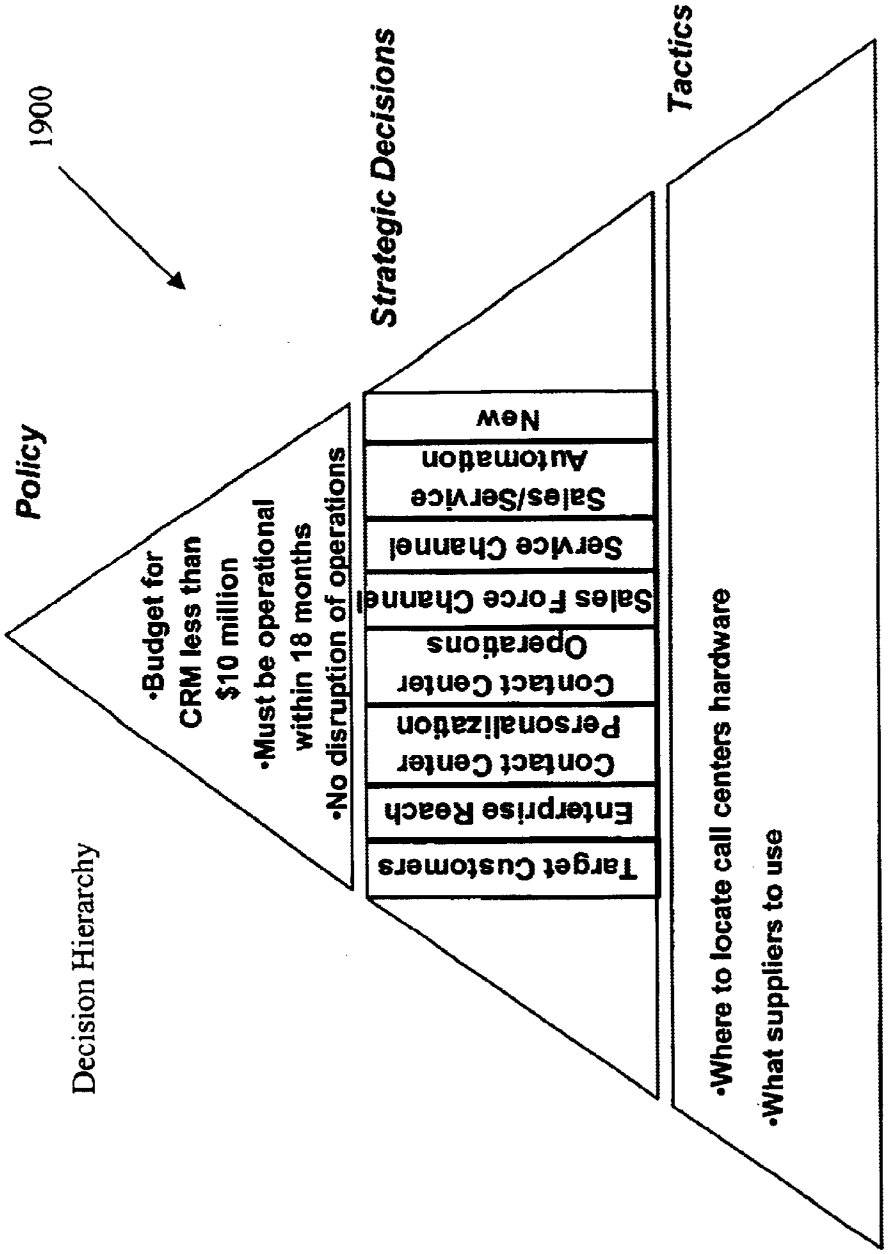

1900
Decision Hierarchy
Policy
•Budget for CRM less than $10 million
•Must be operational within 18 months
•No disruption of operations
Strategic Decisions
Target Customers
Enterprise Reach
Contact Center Personalization
Contact Center Operations
Sales Force Channel
Service Channel
Sales/Service Automation
New
Tactics
•Where to locate call centers hardware
•What suppliers to use

Fig. 19

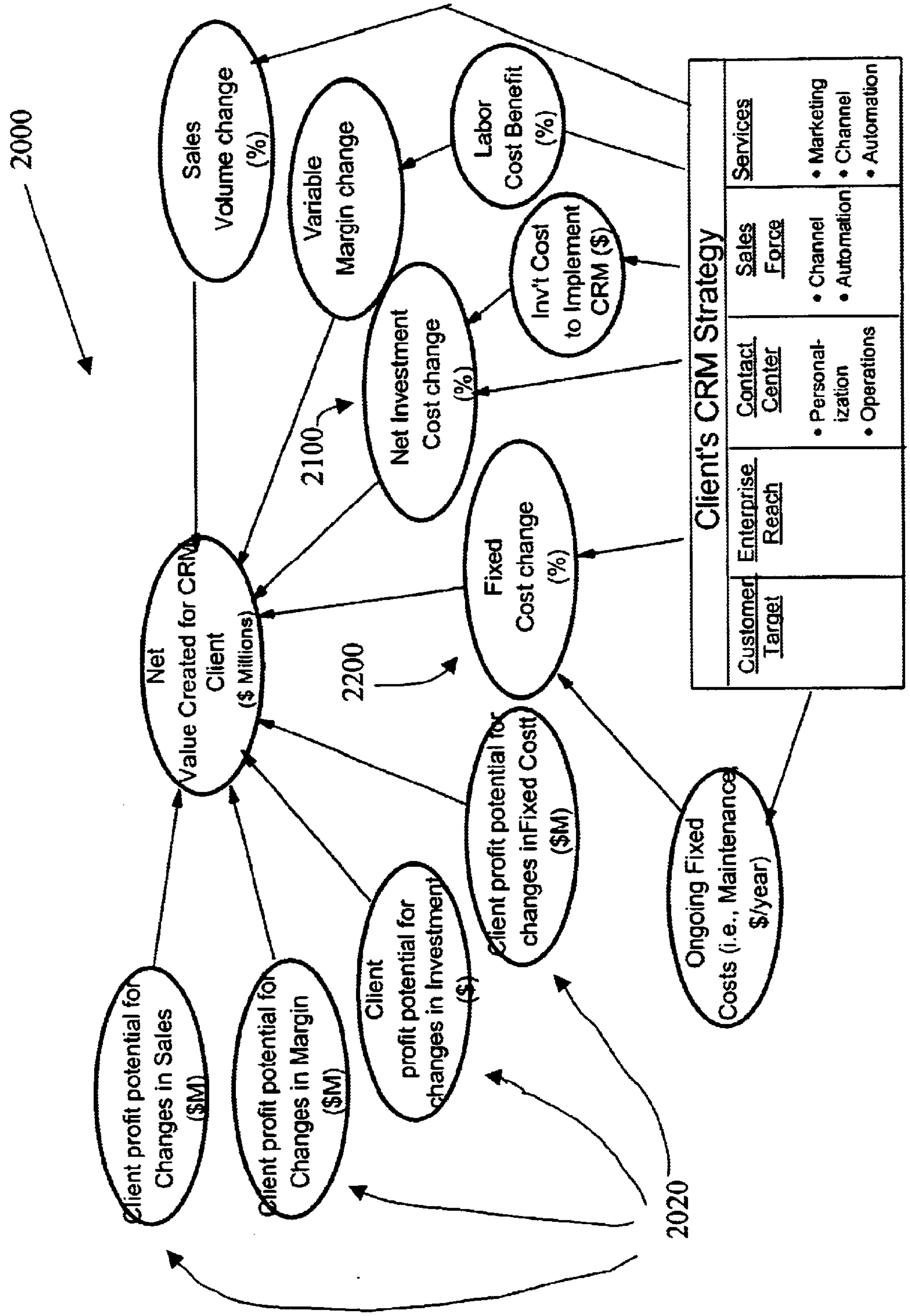

2000
Sales Volume change (%)
Variable Margin change
Labor Cost Benefit (%)
Inv't Cost to Implement CRM ($)
Net Investment Cost change (%)
2100
Fixed Cost change (%)
2200
Net Value Created for CRM Client ($ Millions)
Client's CRM Strategy
Customer Target
Enterprise Reach
Contact Center
• Personal-ization
• Operations
Sales Force
• Channel
• Automation
Services
• Marketing
• Channel
• Automation
Client profit potential for changes inFixed Costt ($M)
Ongoing Fixed Costs (i.e., Maintenance $/year)
Client profit potential for changes in Investment ($)
Client profit potential for Changes in Margin ($M)
Client profit potential for Changes in Sales ($M)
2020

Fig. 20

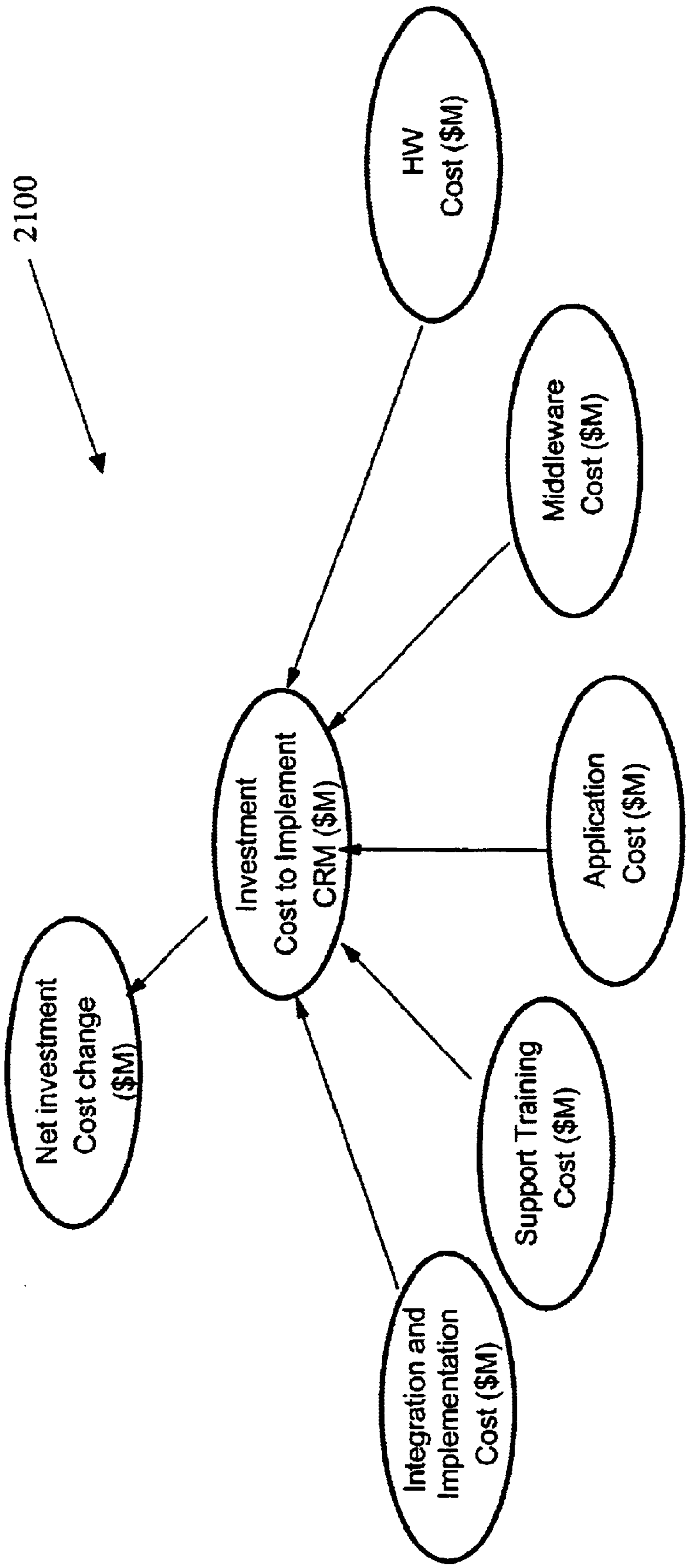

2100
HW Cost ($M)
Middleware Cost ($M)
Application Cost ($M)
Support Training Cost ($M)
Integration and Implementation Cost ($M)
Investment Cost to Implement CRM ($M)
Net investment Cost change ($M)

Fig. 21

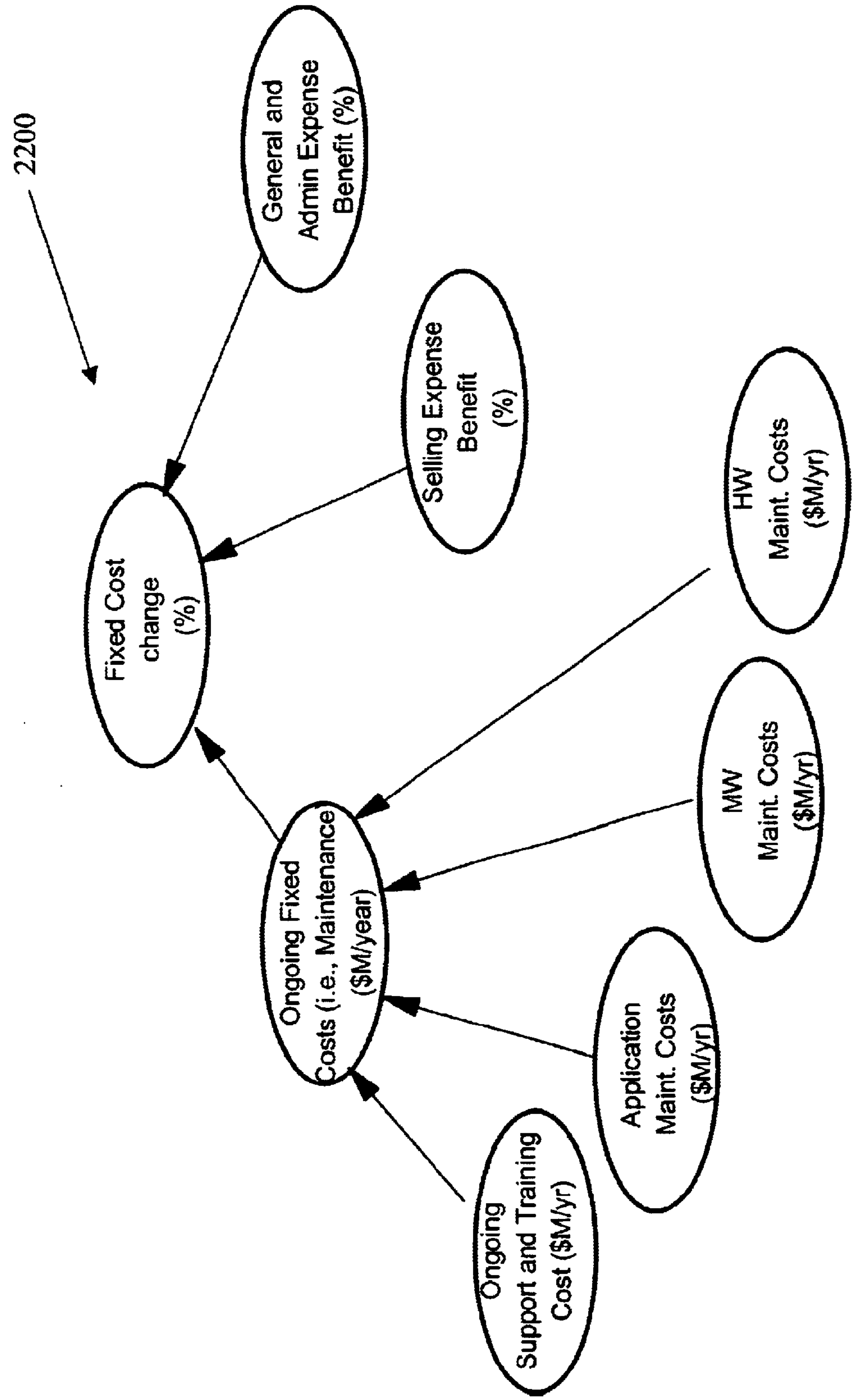

2200
General and
Admin Expense
Benefit (%)
Selling Expense
Benefit
(%)
Fixed Cost
change
(%)
HW
Maint. Costs
($M/yr)
MW
Maint. Costs
($M/yr)
Ongoing Fixed
Costs (i.e., Maintenance
($M/year)
Application
Maint. Costs
($M/yr)
Ongoing
Support and Training
Cost ($M/yr)

Fig. 22

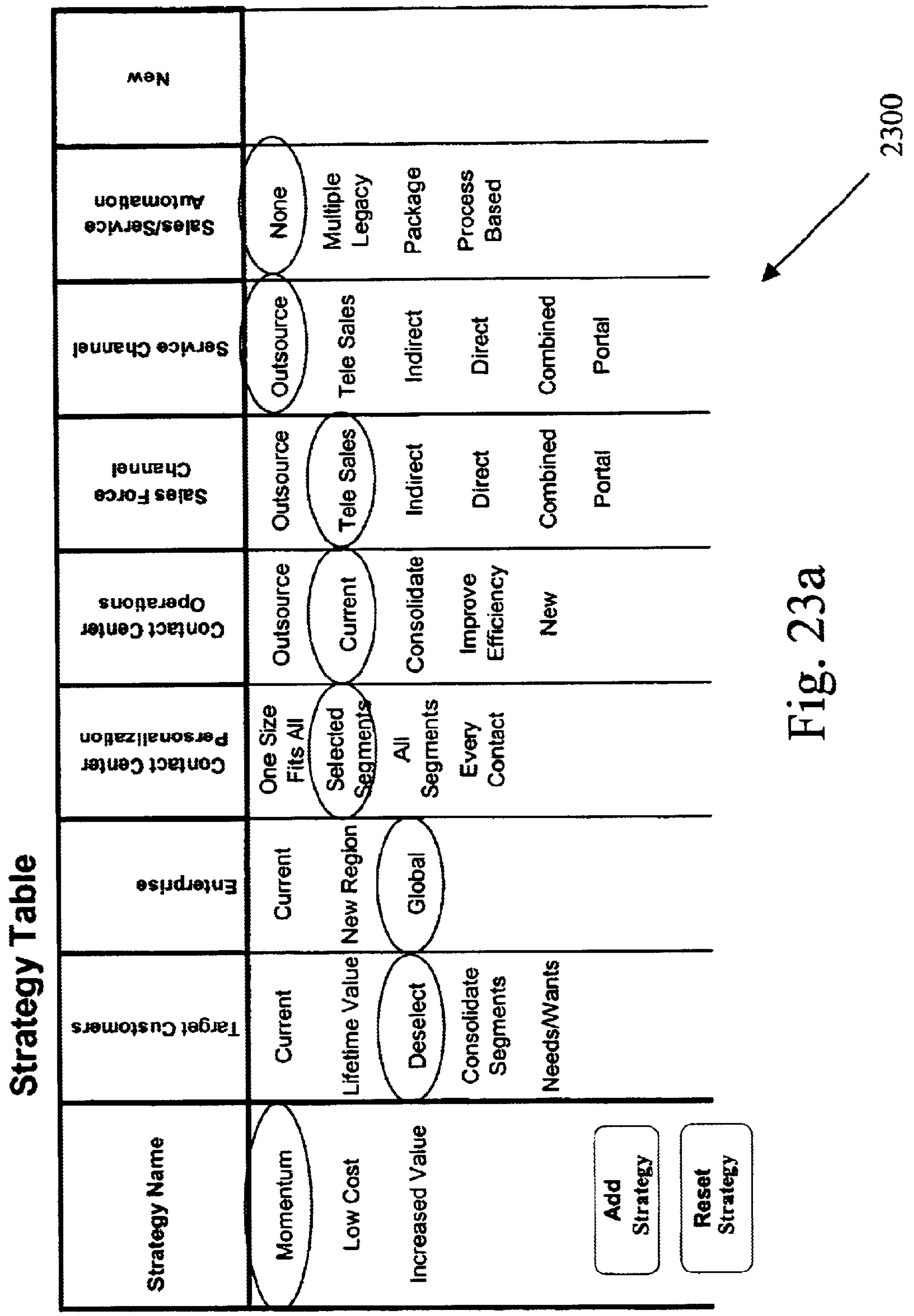

Strategy Table
Strategy Name
Momentum
Low Cost
Increased Value
Add Strategy
Reset Strategy
Target Customers
Current
Lifetime Value
Deselect
Consolidate Segments
Needs/Wants
Enterprise
Current
New Region
Global
Contact Center Personalization
One Size Fits All
Selected Segments
All Segments
Every Contact
Contact Center Operations
Outsource
Current
Consolidate
Improve Efficiency
New
Sales Force Channel
Outsource
Tele Sales
Indirect
Direct
Combined
Portal
Service Channel
Outsource
Tele Sales
Indirect
Direct
Combined
Portal
Sales/Service Automation
None
Multiple Legacy
Package
Process Based
New
2300

Fig. 23a

Strategy Table

Strategy Name	Target Customers	Enterprise Reach	Contact Center Personalization	Contact Center Operations	Sales Force Channel	Service Channel	Sales/Service Automation	New
Momentum	Current	Current	One Size Fits All	Outsource	Outsource	Outsource	None	
Low Cost	Lifetime Value	New Region	Selected Segments	Current	Tele Sales	Tele Sales	Multiple Legacy	
Increased Value	Deselect	Global	All Segments	Consolidate	Indirect	Indirect	Package	
	Consolidate Segments		Every Contact	Improve Efficiency	Direct	Direct	Process Based	
	Needs/Wants			New	Combined	Combined		
					Portal	Portal		

Add Strategy

Reset Strategy

2302

Fig. 23b

Strategy Table

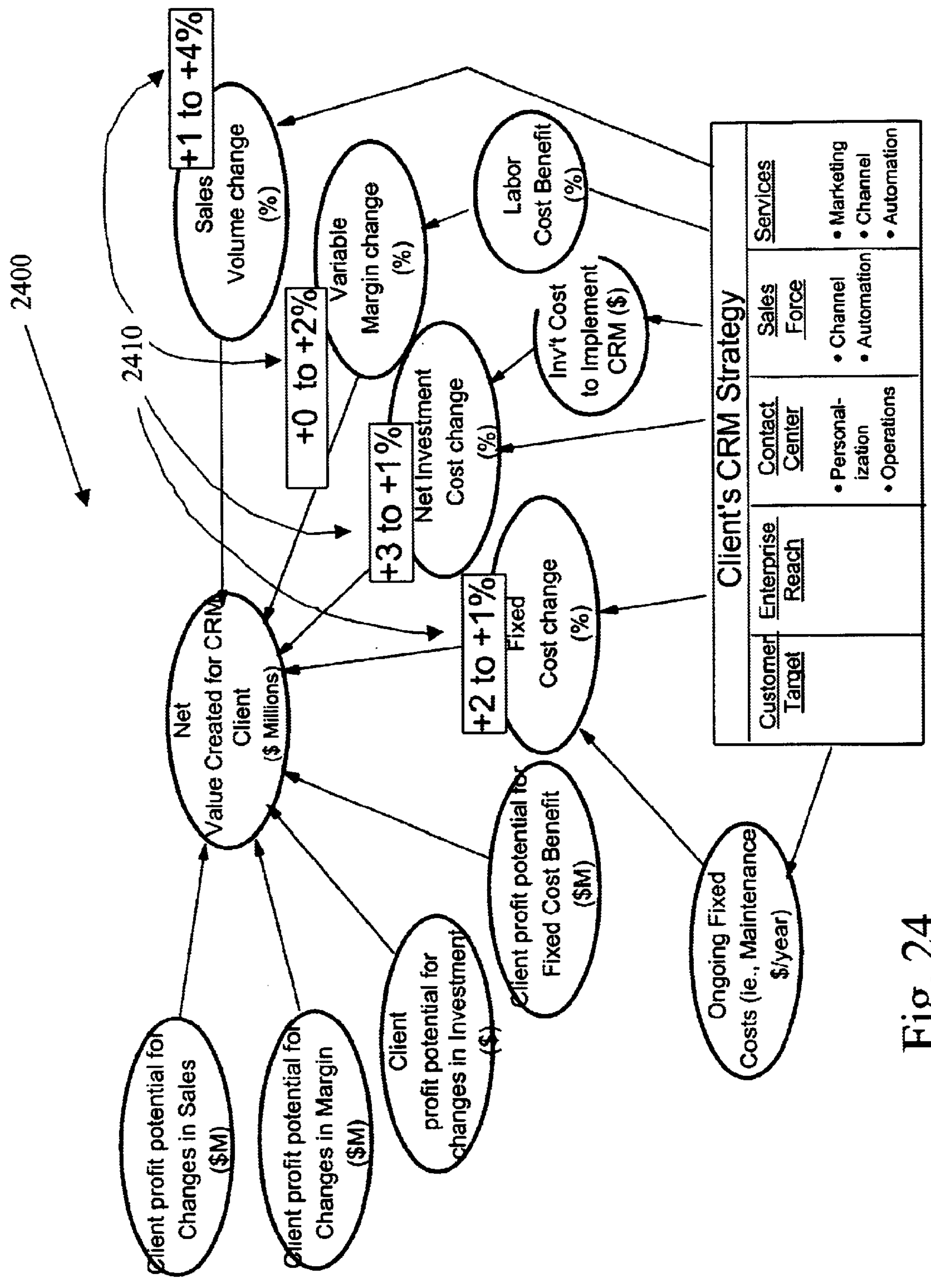
Strategy Name	Target Customers	Enterprise Reach	Contact Center Personalization	Contact Center Operations	Sales Force Channel	Service Channel	Sales/Service Automation	New
Momentum	Current	Current	One Size Fits All	Outsource	Outsource	Outsource	None	
Low Cost	Lifetime Value	New Region	Selected Segments	Current	Tele Sales	Tele Sales	Multiple Legacy	
Increased Value	Deselect	Global	All Segments	Consolidate	Indirect	Indirect	Package	
	Consolidate Segments		Every Contact	Improve Efficiency	Direct	Direct	Process Based	
	Needs/Wants			New	Combined	Combined		
					Portal	Portal		

Add Strategy

Reset Strategy

2304

Fig. 23c



2400
2410
+1 to +4%
Sales Volume change (%)
Variable Margin change (%)
+0 to +2%
Labor Cost Benefit (%)
Inv't Cost to Implement CRM ($)
Net Investment Cost change (%)
+3 to +1%
Fixed Cost change (%)
+2 to +1%
Net Value Created for CRM Client ($ Millions)
Client's CRM Strategy
Customer Target
Enterprise Reach
Contact Center
• Personal-ization
• Operations
Sales Force
• Channel
• Automation
Services
• Marketing
• Channel
• Automation
Client profit potential for Fixed Cost Benefit ($M)
Ongoing Fixed Costs (ie., Maintenance $/year)
Client profit potential for changes in Investment ($)
Client profit potential for Changes in Margin ($M)
Client profit potential for Changes in Sales ($M)

Fig. 24

C3 Attribute	Value of a one percent increase in the C3 attribute **Typical New Vehicle for Automotive manufacturer ($ Millions/year)**	Value of a one percent increase in the C3 attribute **Typical new software product for technology company ($ Millions/year)**
Sales Volume (units)	100	40
Variable Margin (%)	170	15
Investment Cost ($)	-20	-3
Fixed Cost ($)	-10	-30

2500

Fig. 25

C3 Attribute	Value of a one percent increase in the C3 attribute for a typical New Vehicle for Automotive manufacturer ($ Millions/year	Alternative 3: Best Buy/ Cost -Driven (Percent increase in the C3 attribute)	Total value for C3 attribute increase ($ Millions/year)
Sales Volume (units)	100	+2	200
Variable Margin (%)	170	+1	170
Investment Cost ($)	-20	+2	-40
Fixed Cost ($)	-10	+2	-20
Total			310

2600

Fig. 26

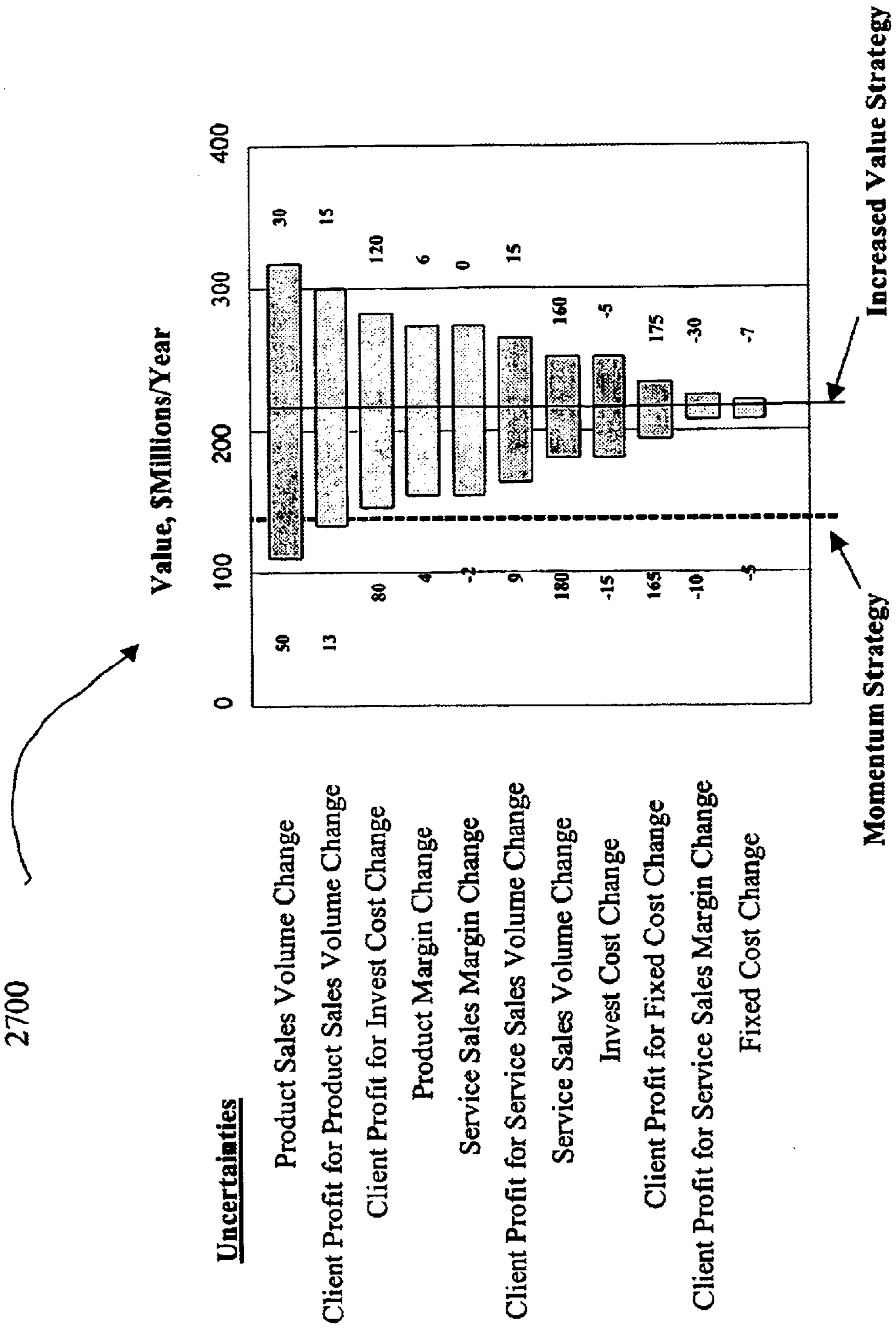

2700
Uncertainties
Product Sales Volume Change
Client Profit for Product Sales Volume Change
Client Profit for Invest Cost Change
Product Margin Change
Service Sales Margin Change
Client Profit for Service Sales Volume Change
Service Sales Volume Change
Invest Cost Change
Client Profit for Fixed Cost Change
Client Profit for Service Sales Margin Change
Fixed Cost Change
Value, $Millions/Year
0
100
200
300
400
50
13
80
4
-2
9
180
-15
165
-10
-5
30
15
120
6
0
15
160
-5
175
-30
-7
Increased Value Strategy
Momentum Strategy

Fig. 27

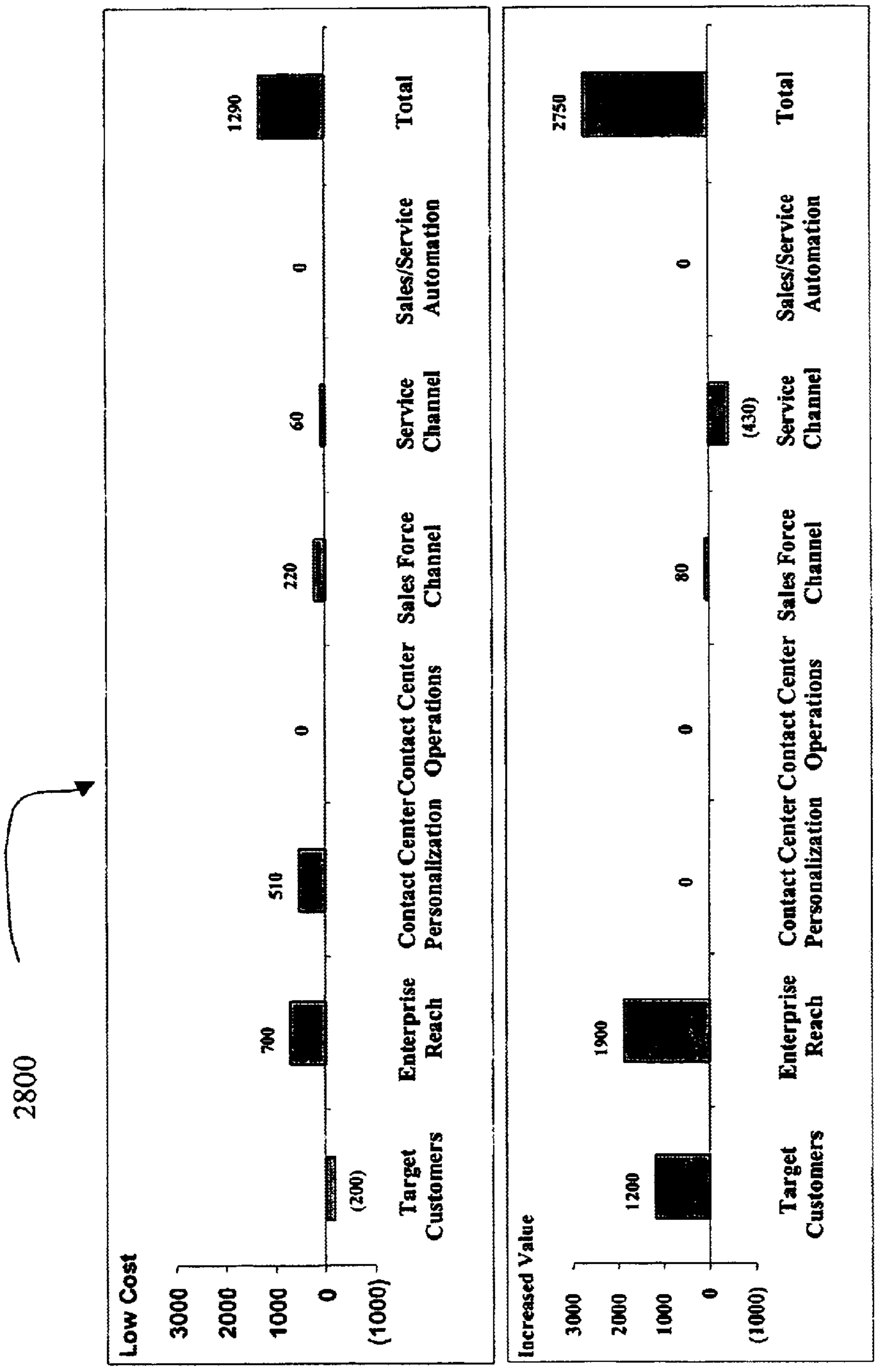

2800
Low Cost
3000
2000
1000
0
(1000)
(200)
700
510
0
220
60
0
1290
Target Customers
Enterprise Reach
Contact Center Personalization
Contact Center Operations
Sales Force Channel
Service Channel
Sales/Service Automation
Total
Increased Value
3000
2000
1000
0
(1000)
1200
1900
0
0
80
(430)
0
2750
Target Customers
Enterprise Reach
Contact Center Personalization
Contact Center Operations
Sales Force Channel
Service Channel
Sales/Service Automation
Total

Fig. 28

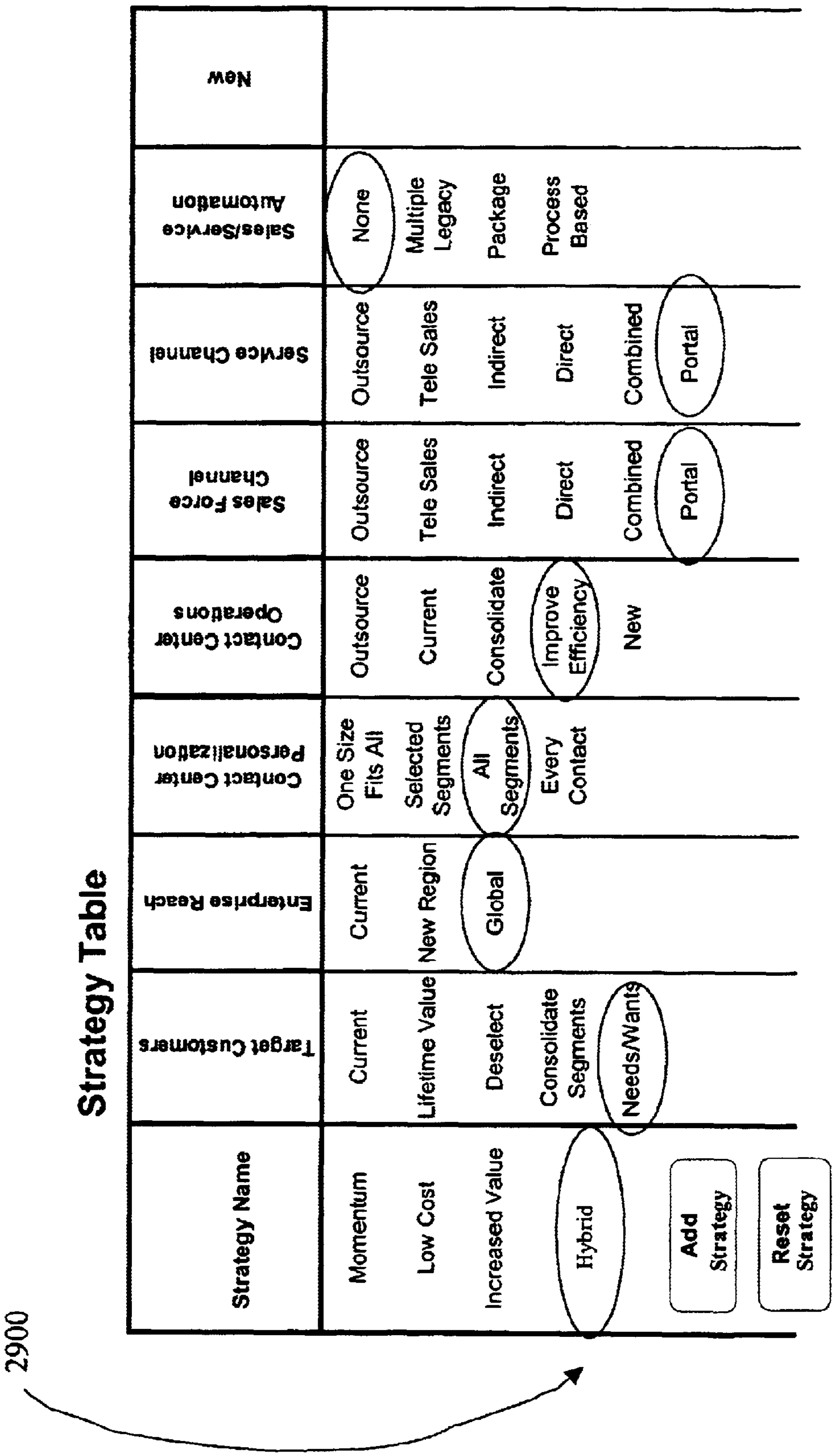
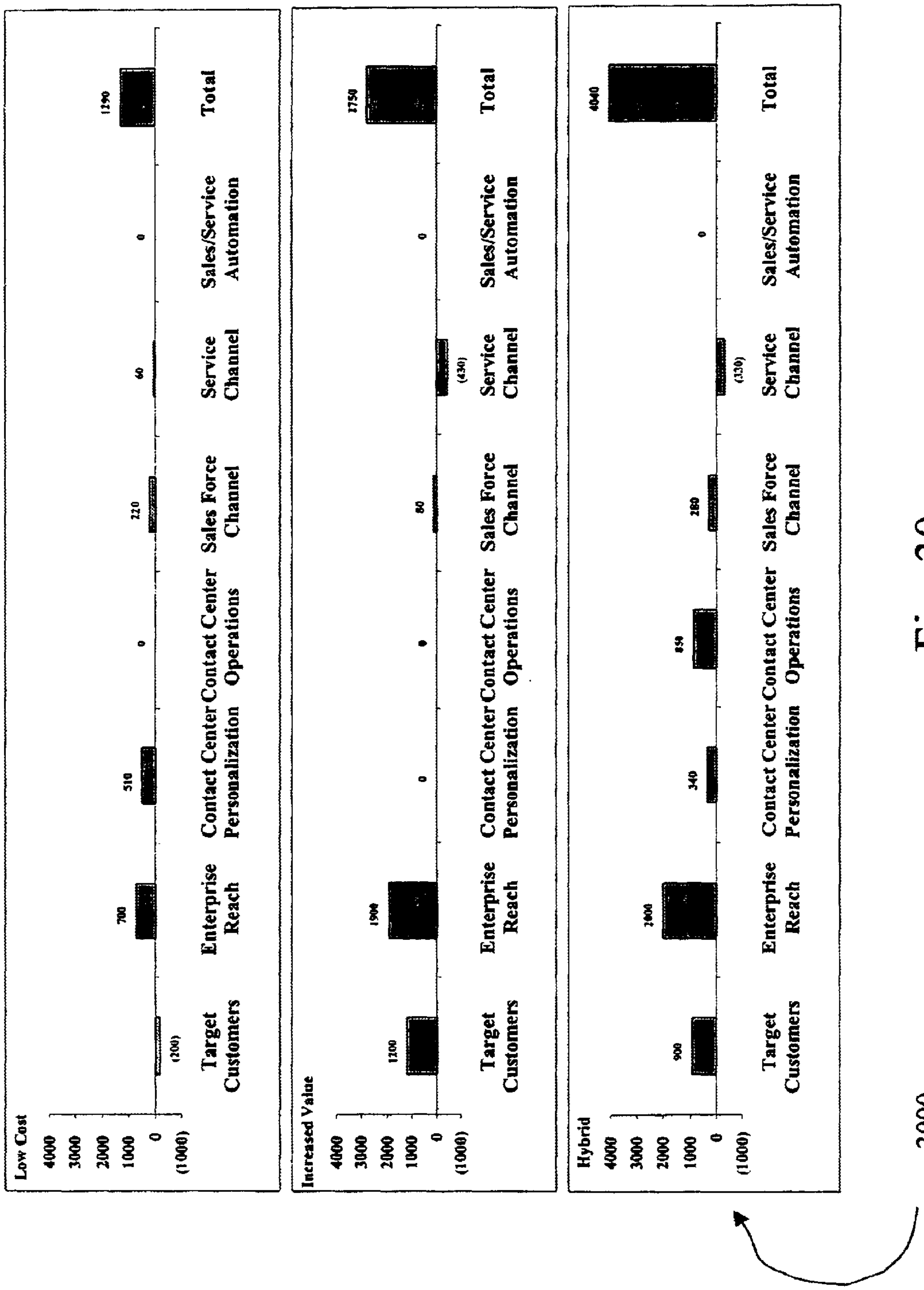

2900
Strategy Table
Strategy Name
Momentum
Low Cost
Increased Value
Hybrid
Add Strategy
Reset Strategy
Target Customers
Current
Lifetime Value
Deselect
Consolidate Segments
Needs/Wants
Enterprise Reach
Current
New Region
Global
Contact Center Personalization
One Size Fits All
Selected Segments
All Segments
Every Contact
Contact Center Operations
Outsource
Current
Consolidate
Improve Efficiency
New
Sales Force Channel
Outsource
Tele Sales
Indirect
Direct
Combined
Portal
Service Channel
Outsource
Tele Sales
Indirect
Direct
Combined
Portal
Sales/Service Automation
None
Multiple Legacy
Package
Process Based
New

Fig. 29


Low Cost
4000
3000
2000
1000
0
(1000)
(200)
700
510
0
220
60
0
1290
Target Customers
Enterprise Reach
Contact Center Personalization
Contact Center Operations
Sales Force Channel
Service Channel
Sales/Service Automation
Total
Increased Value
4000
3000
2000
1000
0
(1000)
1200
1900
0
0
80
(430)
0
1750
Target Customers
Enterprise Reach
Contact Center Personalization
Contact Center Operations
Sales Force Channel
Service Channel
Sales/Service Automation
Total
Hybrid
4000
3000
2000
1000
0
(1000)
900
2000
340
850
280
(330)
0
4040
Target Customers
Enterprise Reach
Contact Center Personalization
Contact Center Operations
Sales Force Channel
Service Channel
Sales/Service Automation
Total

3000

Fig. 30

SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR A CUSTOMER-CENTRIC COLLABORATIVE PROTOCOL

RELATED APPLICATION(S)

The present application claims the priority of a previously filed provisional application with the title "Collaborative Decision Platform" filed Nov. 8, 1999 under Ser. No. 60/163,984, which is incorporated herein by reference in its entirety. The present application is further related to an application filed concurrently herewith under the title "System, Method and Computer Program Product for a Collaborative Decision Platform" by inventors Daniel L. Owen and Michael W. Kusnic under Ser. No. 09/708,154, which is incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

The present invention relates to decision-making logic, and more particularly to a computer-based platform that supports a decision making process and business-to-business exchanges.

BACKGROUND OF THE INVENTION

Historically, enterprises have used "Make and Sell" to define their products and services. Prior art FIG. **1***a* illustrates an example of the make and sell approach. As shown, a technical or marketing organization **10** makes specifications or requirements for an enterprise **12** which, in turn, generates a product or service for being sold to a consumer **14**. The technical or marketing organization **10** may produce the specifications or requirements based on information about the customer, shown. Within the enterprise there may be a formal structure for collection of information about customers, but that structure is not common or "scalable" across enterprises.

A new model of collaborative decision making, which is "Market-based", has proven very effective for delivering more valuable products and services to consumers. FIG. **1***b* illustrates an example of a market-based approach of the prior art. As shown, an intermediate collaborative decision making process **16** receives the value or needs of the consumer **14**. Also, the enterprise **12** provides such intermediate collaborative decision making process **16** with product/service alternatives based on the capabilities of its technical or marketing organizations **10**. The intermediate collaborative decision making process **16** in turn contributes in forming an agreement among the decision-makers within the enterprise **12** to produce high-value products and services for the consumer.

While this model has been proven for consumer-to-business frameworks, no work has been done to extend this model to business-to-business (B2B) frameworks in a manner that operates effectively. In a B2B framework, the customer is another business rather than a consumer. There is therefore a need for a computer implemented business method for providing a consumer centric collaborative protocol.

DISCLOSURE OF THE INVENTION

A system, method and computer program product are provided for affording customer-centric collaborative decision making in a business-to-business framework. Initially, a minimum set of attributes is defined. Thereafter, first information regarding each of the minimum set of attributes is received from a receiving business. Second information is

then received regarding proposed products or services in terms of the minimum set of attributes. Such second information is received from a supplying business. In use, a decision process is executed based on the first information and the second information as to which products or services most valuable to the receiving business.

In one embodiment of the present invention, the attributes include price, sales, variable costs, fixed cost, and investment. Further, the attributes may also include market share, market size, labor cost, material cost, administrative cost, annual expenses, working capital, planning and equipment, etc. As an option, the first information and second information may be received utilizing a network. Such network may take the form of the Internet.

In still another embodiment of the present invention, the foregoing capabilities may be carried out using a collaborative decision platform adapted to run on a computer. In such system, an application capable of performing decision logic is executed. Information is then retrieved from a database in accordance with the decision logic. Information is then provided to and received from users in accordance with the decision logic utilizing a user interface. The information is then processed utilizing the decision logic. In use, the foregoing steps are carried out by a collaborative decision platform capable of retrieving and receiving the information, and processing such information for different purposes by executing different applications each capable of performing different decision logic. It should be noted that the various steps set forth hereinabove may be carried out using universal modules capable of interfacing with different applications.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. **1***a* and **1***b* illustrate prior art systems;

FIG. **1***c* illustrates a method for providing a collaborative decision platform adapted to run on a computer,

FIG. **1***d* illustrates a system by which the method of FIG. **1***c* may be carried out;

FIG. **1***e* illustrates a networked decision making environment in accordance with one embodiment of the present invention;

FIG. **2** shows a representative hardware environment on which the collaborative decision platform of FIG. **1***d* may be implemented;

FIG. **3** illustrates an example of Framing in accordance with one embodiment of the present invention;

FIG. **3***a* illustrates various logic associated with the Framing process of the present invention;

FIG. **4** illustrates an example of Alternatives in accordance with one embodiment of the present invention;

FIG. **4***a* illustrates various logic associated with the Alternatives process of the present invention which is capable of handling its various input for the purpose of generating a strategy table;

FIG. **5** illustrates an example of Analysis in accordance with one embodiment of the present invention;

FIG. **5***a* illustrates various logic associated with the Analysis process of the present invention;

FIG. **6** illustrates an example of Connection in accordance with one embodiment of the present invention;

FIG. **6***a* illustrates various logic associated with the Connection process of the present invention;

FIG. **7** illustrates the various logical connectivity between the various inputs and outputs of the Framing, Alternatives, Analysis, and Connection logic that comprises the users' interface;

FIGS. **8***a-i* illustrate an example of an application of the various logic components set forth in FIGS. **3–7**;

FIG. **9** illustrates a method for affording customer-centric collaborative decision making in a business-to-business framework;

FIGS. **9***a* and **10** illustrates tables associated with the method of FIG. **9**;

FIG. **11** is a schematic diagram showing the customer-centric collaborative protocol;

FIG. **12** illustrates a first example of the embodiment set forth in FIG. **11**;

FIG. **13** illustrates a second example of the embodiment set forth in FIG. **11**;

FIGS. **14** and **15** illustrate third and fourth examples, respectively, of the embodiment set forth in FIG. **11**, where an industry independent, open and scalable platform is provided for business-to business exchange of existing goods and services that are not commodities;

FIG. **16** illustrates a fifth example of the embodiment set forth in FIG. **11**, where an industry independent, open and scalable platform is provided for B2B real-time collaboration in the definition of future, non-existent goods and services;

FIGS. **17** and **18** illustrate sixth and seventh examples, respectively, of the embodiment set forth in FIG. **11**, where a new business design is provided that assists business-to business enterprises in measuring the value creation for its customers; and

FIGS. **19** through **30** illustrate an exemplary application of the customer centric collaborative protocol.

DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. **1***c* illustrates a method **100** for providing a collaborative decision platform adapted to run on a computer. Initially, an application capable of performing decision logic is executed. See operation **102**.

Information is then retrieved from a database in accordance with the decision logic, as indicated in operation **104**. Information is then delivered to and received from a user in accordance with the decision logic utilizing a user interface. Note operation **106**. The information is then processed in operation **108** utilizing the decision logic.

In use, the foregoing steps are carried out by a collaborative decision platform capable of retrieving and receiving the information, and processing such information for different purposes by executing different applications each capable of performing different decision logic. Note operation **110**. It should be noted that the various steps set forth hereinabove may be carried out using universal modules capable of interfacing with different applications.

FIG. **1***d* illustrates a system **120** by which the foregoing method of FIG. **1***c* may be carried out. As shown, a collaborative decision platform **122** is provided which has an interface **125** with at least one application **124** for executing the decision logic, as set forth in operation **102** of FIG. **1***c*. Further included is a database **126**, which has an interface **127** with the collaborative decision platform **122** in accordance with operation **104** of FIG. **1***c*. Further, a user interface **128** is provided for receiving information from and providing information to the users. The interfaces **125**, **127**, and **128** are defined by the collaborative decision platform **122**. The users may be an important element of the system **120**. Note the two-headed arrow representing the users' interface **128** with the collaborative decision platform **122** to indicate the interaction, while the single arrowhead of the interface **125** and **127** indicates input. Note operation **106** of FIG. **1***c*. The collaborative decision platform **122** may be run on any type of hardware architecture **130**.

As set forth earlier, the various steps of FIG. **1***c* may be carried out using universal modules capable of interfacing with different applications. Such different applications **124** may be capable of performing decision logic relating to any type of decision-making process (e.g. financial, medical, buying a house, selecting a corporate strategy, etc.). In use, the collaborative decision platform **122** enables decision-making processes through the sequence and connectivity of a set of common displays, which describes the decision to be made. The collaborative decision platform **122** further enables asynchronous, remote decision-making processes, i.e. the ability to have different people input data into the set of common displays at different times, and from different places. Further, the database **126** may take the form of any one or a plurality of databases which may or may not be interconnected via a network such as the Internet. To this end, the present embodiment is designed to foster clear and conscientious decision-making.

FIG. **1***e* illustrates a plurality of network **130** of decision environments for allowing enterprises to learn more rapidly and coordinate more effectively. Such a network of decision environments each include at least one collaborative user interface which each communicate with an enterprise learning and coordination module **132** that may include one or more collaborative decision platforms **122**. Such a network **130** may allow the decision environments to be a physical arrangement optimized for human decision making or a virtual environment consisting of only the computer hardware and the collaborative decision platform **122**.

FIG. **2** shows a representative hardware environment on which the collaborative decision platform **122** of FIG. **1***d* may be implemented. Such figure illustrates a typical hardware configuration of a workstation in accordance with a preferred embodiment having a central processing unit **210**, such as a microprocessor, and a number of other units interconnected via a system bus **212**.

The workstation shown in FIG. **2** includes a Random Access Memory (RAM) **214**, Read Only Memory (ROM) **216**, an I/O adapter **218** for connecting peripheral devices such as disk storage units **220** to the bus **212**, a user interface adapter **222** for connecting a keyboard **224**, a mouse **226**, a speaker **228**, a microphone **232**, and/or other user interface devices such as a touch screen (not shown) to the bus **212**, communication adapter **234** for connecting the workstation to a communication network **235** (e.g., a data processing network) and a display adapter **236** for connecting the bus **212** to a display device **238**.

The workstation typically has resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating System (OS), the IBM OS/2 operating system, the MAC OS, or UNIX operating system. Those skilled in the art will appreciate that the present invention may also be implemented on platforms and operating systems other than those mentioned.

A preferred embodiment is written using JAVA, C, and the C++ language and utilizes object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications. As OOP moves toward the mainstream of software design and development, various software solutions require adaptation to make use of the benefits of OOP. A need exists for these principles of OOP to be applied to a messaging interface of an electronic messaging system such that a set of OOP classes and objects for the messaging interface can be provided.

OOP is a process of developing computer software using objects, including the steps of analyzing the problem, designing the system, and constructing the program. An object is a software package that contains both data and a collection of related structures and procedures. Since it contains both data and a collection of structures and procedures, it can be visualized as a self-sufficient component that does not require other additional structures, procedures or data to perform its specific task. OOP, therefore, views a computer program as a collection of largely autonomous components, called objects, each of which is responsible for a specific task. This concept of packaging data, structures, and procedures together in one component or module is called encapsulation.

In general, OOP components are reusable software modules which present an interface that conforms to an object model and which are accessed at run-time through a component integration architecture. A component integration architecture is a set of architecture mechanisms which allow software modules in different process spaces to utilize each others capabilities or functions. This is generally done by assuming a common component object model on which to build the architecture. It is worthwhile to differentiate between an object and a class of objects at this point. An object is a single instance of the class of objects, which is often just called a class. A class of objects can be viewed as a blueprint, from which many objects can be formed.

OOP allows the programmer to create an object that is a part of another object. For example, the object representing a piston engine is said to have a composition-relationship with the object representing a piston. In reality, a piston engine comprises a piston, valves and many other components; the fact that a piston is an element of a piston engine can be logically and semantically represented in OOP by two objects.

OOP also allows creation of an object that "depends from" another object. If there are two objects, one representing a piston engine and the other representing a piston engine wherein the piston is made of ceramic, then the relationship between the two objects is not that of composition. A ceramic piston engine does not make up a piston engine. Rather it is merely one kind of piston engine that has one more limitation than the piston engine; its piston is made of ceramic. In this case, the object representing the ceramic piston engine is called a derived object, and it inherits all of the aspects of the object representing the piston engine and adds further limitation or detail to it. The object representing the ceramic piston engine "depends from" the object representing the piston engine. The relationship between these objects is called inheritance.

When the object or class representing the ceramic piston engine inherits all of the aspects of the objects representing the piston engine, it inherits the thermal characteristics of a standard piston defined in the piston engine class. However, the ceramic piston engine object overrides these ceramic specific thermal characteristics, which are typically different from those associated with a metal piston. It skips over the original and uses new functions related to ceramic pistons. Different kinds of piston engines have different characteristics, but may have the same underlying functions associated with it (e.g., how many pistons in the engine, ignition sequences, lubrication, etc.). To access each of these functions in any piston engine object, a programmer would call the same functions with the same names, but each type of piston engine may have different/overriding implementations of functions behind the same name. This ability to hide different implementations of a function behind the same name is called polymorphism and it greatly simplifies communication among objects.

With the concepts of composition-relationship, encapsulation, inheritance and polymorphism, an object can represent just about anything in the real world. In fact, one's logical perception of the reality is the only limit on determining the kinds of things that can become objects in object-oriented software. Some typical categories are as follows:

Objects can represent physical objects, such as automobiles in a traffic-flow simulation, electrical components in a circuit-design program, countries in an economics model, or aircraft in an air-traffic-control system.

Objects can represent elements of the computer-user environment such as windows, menus or graphics objects.

An object can represent an inventory, such as a personnel file or a table of the latitudes and longitudes of cities.

An object can represent user-defined data types such as time, angles, and complex numbers, or points on the plane.

With this enormous capability of an object to represent just about any logically separable matters, OOP allows the software developer to design and implement a computer program that is a model of some aspects of reality, whether that reality is a physical entity, a process, a system, or a composition of matter. Since the object can represent anything, the software developer can create an object which can be used as a component in a larger software project in the future.

If 90% of a new OOP software program consists of proven, existing components made from preexisting reusable objects, then only the remaining 10% of the new software project has to be written and tested from scratch. Since 90% already came from an inventory of extensively tested reusable objects, the potential domain from which an error could originate is 10% of the program. As a result, OOP enables software developers to build objects out of other, previously built objects.

This process closely resembles complex machinery being built out of assemblies and sub-assemblies. OOP technology, therefore, makes software engineering more like hardware engineering in that software is built from existing components, which are available to the developer as objects. All this adds up to an improved quality of the software as well as an increased speed of its development.

Programming languages are beginning to fully support the OOP principles, such as encapsulation, inheritance, polymorphism, and composition-relationship. With the advent of the C++ language, many commercial software developers have embraced OOP. C++ is an OOP language that offers a fast, machine-executable code. Furthermore, C++ is suitable for both commercial-application and systems-programming projects. For now, C++ appears to be the most popular choice among many OOP programmers, but there is a host of other OOP languages, such as Smalltalk, Common Lisp Object System (CLOS), and Eiffel. Additionally, OOP capabilities are being added to more traditional popular computer programming languages such as Pascal.

The benefits of object classes can be summarized, as follows:

Objects and their corresponding classes break down complex programming problems into many smaller, simpler problems.

Encapsulation enforces data abstraction through the organization of data into small, independent objects that can communicate with each other. Encapsulation protects the data in an object from accidental damage, but allows other objects to interact with that data by calling the object's member functions and structures.

Subclassing and inheritance make it possible to extend and modify objects through deriving new kinds of objects from the standard classes available in the system. Thus, new capabilities are created without having to start from scratch.

Polymorphism and multiple inheritance make it possible for different programmers to mix and match characteristics of many different classes and create specialized objects that can still work with related objects in predictable ways.

Class hierarchies and containment hierarchies provide a flexible mechanism for modeling real-world objects and the relationships among them.

Libraries of reusable classes are useful in many situations, but they also have some limitations. For example:

Complexity. In a complex system, the class hierarchies for related classes can become extremely confusing, with many dozens or even hundreds of classes.

Flow of control. A program written with the aid of class libraries is still responsible for the flow of control (i.e., it must control the interactions among all the objects created from a particular library). The programmer has to decide which functions to call at what times for which kinds of objects.

Duplication of effort. Although class libraries allow programmers to use and reuse many small pieces of code, each programmer puts those pieces together in a different way. Two different programmers can use the same set of class libraries to write two programs that do exactly the same thing but whose internal structure (i.e., design) may be quite different, depending on hundreds of small decisions each programmer makes along the way. Inevitably, similar pieces of code end up doing similar things in slightly different ways and do not work as well together as they should.

Class libraries are very flexible. As programs grow more complex, more programmers are forced to reinvent basic solutions to basic problems over and over again. A relatively new extension of the class library concept is to have a framework of class libraries. This framework is more complex and consists of significant collections of collaborating classes that capture both the small scale patterns and major mechanisms that implement the common requirements and design in a specific application domain. They were first developed to free application programmers from the chores involved in displaying menus, windows, dialog boxes, and other standard user interface elements for personal computers.

Frameworks also represent a change in the way programmers think about the interaction between the code they write and code written by others. In the early days procedural programming, the programmer called libraries provided by the operating system to perform certain tasks, but basically the program executed down the page from start to finish, and the programmer was solely responsible for the flow of control. This was appropriate for printing out paychecks, calculating a mathematical table, or solving other problems with a program that executed in just one way.

The development of graphical user interfaces began to turn this procedural programming arrangement inside out. These interfaces allow the user, rather than program logic, to drive the program and decide when certain actions should be performed. Today, most personal computer software accomplishes this by means of an event loop which monitors the mouse, keyboard, and other sources of external events and calls the appropriate parts of the programmer's code according to actions that the user performs. The programmer no longer determines the order in which events occur. Instead, a program is divided into separate pieces that are called at unpredictable times and in an unpredictable order. By relinquishing control in this way to users, the developer creates a program that is much easier to use. Nevertheless, individual pieces of the program written by the developer still call libraries provided by the operating system to accomplish certain tasks, and the programmer must still determine the flow of control within each piece after it's called by the event loop. Application code still "sits on top of" the system.

Even event loop programs require programmers to write a lot of code that should not need to be written separately for every application. The concept of an application framework carries the event loop concept further. Instead of dealing with all the nuts and bolts of constructing basic menus, windows, and dialog boxes and then making these things all work together, programmers using application frameworks start with working application code and basic user interface elements in place. Subsequently, they build from there by replacing some of the generic capabilities of the framework with the specific capabilities of the intended application.

Application frameworks reduce the total amount of code that a programmer has to write from scratch. However, because the framework is really a generic application that displays windows, supports copy and paste, and so on, the programmer can also relinquish control to a greater degree than event loop programs permit. The framework code takes care of almost all event handling and flow of control, and the programmer's code is called only when the framework needs it (e.g., to create or manipulate a proprietary data structure).

A programmer writing a framework program not only relinquishes control to the user (as is also true for event loop programs), but also relinquishes the detailed flow of control within the program to the framework. This approach allows the creation of more complex systems that work together in interesting ways, as opposed to isolated programs, having custom code, being created over and over again for similar problems.

Thus, as is explained above, a framework basically is a collection of cooperating classes that make up a reusable design solution for a given problem domain. It typically includes objects that provide default behavior (e.g., for menus and windows), and programmers use it by inheriting some of that default behavior and overriding other behavior so that the framework calls application code at the appropriate times.

There are three main differences between frameworks and class libraries:

Behavior versus protocol. Class libraries are essentially collections of behaviors that you can call when you want those individual behaviors in your program. A framework, on the other hand, provides not only behavior but also the protocol or set of rules that govern the ways in which behaviors can be combined, including rules for what a programmer is supposed to provide versus what the framework provides.

Call versus override. With a class library, the code the programmer instantiates objects and calls their member functions. It's possible to instantiate and call objects in the same way with a framework (i.e., to treat the framework as a class library), but to take full advantage of a framework's reusable design, a programmer typically writes code that overrides and is called by the framework. The framework manages the flow of control among its objects. Writing a program involves dividing responsibilities among the various pieces of software that are called by the framework rather than specifying how the different pieces should work together.

Implementation versus design. With class libraries, programmers reuse only implementations, whereas with frameworks, they reuse design. A framework embodies the way a family of related programs or pieces of software work. It represents a generic design solution that can be adapted to a variety of specific problems in a given domain. For example, a single framework can embody the way a user interface works, even though two different user interfaces created with the same framework might solve quite different interface problems.

Thus, through the development of frameworks for solutions to various problems and programming tasks, significant reductions in the design and development effort for software can be achieved. A preferred embodiment of the invention utilizes HyperText Markup Language (HTML) to implement documents on the Internet together with a general-purpose secure communication protocol for a transport medium between the client and the Newco. HTTP or other protocols could be readily substituted for HTML without undue experimentation. Information on these products is available in T. Berners-Lee, D. Connoly, "RFC 1866: Hypertext Markup Language—2.0" (November 1995); and R. Fielding, H, Frystyk, T. Berners-Lee, J. Gettys and J. C. Mogul, "Hypertext Transfer Protocol—HTTP/1.1: HTTP Working Group Internet Draft" (May 2, 1996). HTML is a simple data format used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of domains. HTML has been in use by the World-Wide Web global information initiative since 1990. HTML is an application of ISO Standard 8879; 1986 Information Processing Text and Office Systems; Standard Generalized Markup Language (SGML).

To date, Web development tools have been limited in their ability to create dynamic Web applications which span from client to server and interoperate with existing computing resources. Until recently, HTML has been the dominant technology used in development of Web-based solutions. However, HTML has proven to be inadequate in the following areas:

Poor performance;

Restricted user interface capabilities;

Can only produce static Web pages;

Lack of interoperability with existing applications and data; and

Inability to scale.

Sun Microsystem's Java language solves many of the client-side problems by:

Improving performance on the client side;

Enabling the creation of dynamic, real-time Web applications; and

Providing the ability to create a wide variety of user interface components.

With Java, developers can create robust User Interface (UI) components. Custom "widgets" (e.g., real-time stock tickers, animated icons, etc.) can be created, and client-side performance is improved. Unlike HTML, Java supports the notion of client-side validation, offloading appropriate processing onto the client for improved performance. Dynamic, real-time Web pages can be created. Using the above-mentioned custom UI components, dynamic Web pages can also be created.

Sun's Java language has emerged as an industry-recognized language for "programming the Internet." Sun defines Java as: "a simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, dynamic, buzzword-compliant, general-purpose programming language. Java supports programming for the Internet in the form of platform-independent Java applets." Java applets are small, specialized applications that comply with Sun's Java Application Programming Interface (API) allowing developers to add "interactive content" to Web documents (e.g., simple animations, page adornments, basic games, etc.). Applets execute within a Java-compatible browser (e.g., Netscape Navigator) by copying code from the server to client. From a language standpoint, Java's core feature set is based on C++. Sun's Java literature states that Java is basically, "C++ with extensions from Objective C for more dynamic method resolution."

Another technology that provides similar function to JAVA is provided by Microsoft and ActiveX Technologies, to give developers and Web designers wherewithal to build dynamic content for the Internet and personal computers. ActiveX includes tools for developing animation, 3-D virtual reality, video and other multimedia content. The tools use Internet standards, work on multiple platforms, and are being supported by over 100 companies. The group's building blocks are called ActiveX Controls, small, fast components that enable developers to embed parts of software in hypertext markup language (HTML) pages. ActiveX Controls work with a variety of programming languages including Microsoft Visual C++, Borland Delphi, Microsoft Visual Basic programming system and, in the future, Microsoft's development tool for Java, code named "Jakarta." ActiveX Technologies also includes ActiveX Server Framework, allowing developers to create server applications. One of ordinary skill in the art readily recognizes that ActiveX could be substituted for JAVA without undue experimentation to practice the invention.

It should be noted that, in one embodiment, the information database and the common displays may all be treated as objects by the platform. As such, the foregoing technology may be utilized in the implementation of the overall system, as embodied in FIG. 1*d*.

Preferred Embodiment

The platform of the present embodiment acts as a "decision engine" which drives the decision process through a sequence of logical steps to a conclusion. The users' interface during these steps is the set of common displays exhibited by the platform. The users receive and provide specific decision information to the platform by entering or modifying the structure of the decision and the decision-relevant information in the display areas where appropriate. In order to start the process, the platform hosts a decision application which provides the structure for the type of decision that the user wants to make. The application and platform communicate through a standard interface protocol. The platform guides the user through four steps (framing, alternatives, analysis and connection), but these are tailored to the decision at hand through the decision application.

FIG. **3** illustrates an example of Framing **300** in accordance with one embodiment of the present invention. The purpose of Framing is to clearly communicate to the users the capabilities of the chosen decision application **124** and to allow the users to modify the problem definition to the extent that the capability for modification has been incorporated by the authors of the application. During Framing, the specific decision application provides certain key pieces of information about the decision at hand as input in a specific format or protocol **125** specified by the collaborative decision platform **122** that describe the capabilities of that application. Such input may include the policies that form boundary conditions for the decision, the strategic decisions that can be made, the values that are important to the decision makers, the uncertainties that may impact the values desired, and the relationship of the above elements.

The Framing process, using this key input from the decision application **124** in the specific format **125**, generates visual displays of a decision hierarchy **304** and an influence diagram **306**, to be confirmed or modified by the users. The users' information **129** is seen as an input to the framing process **300**, because the users interact with the platform **122** to produce a resultant decision hierarchy **304** and the influence diagram **306** that capture their collective view of the decision problem. Note the two-headed arrow representing the users' interface **128** with the collaborative decision platform **122** to indicate the interaction, while the single arrow head of the interface **125** indicates input. In the event that the users are unable to successfully represent the decision problem as they see it with the initial decision application, they will select another application **124** and repeat the Framing process **300**.

FIG. **3***a* illustrates various logic **310** associated with the Framing process of the present invention. As shown, a first Framing module **314** receives information from the decision application **124**, such as the specific policies, decisions (controllables) and tactics that it can accommodate with a logical structure. The first framing module **314** orders the precedence of decisions to output the decision hierarchy **304**. Decisions that have already been made are referred to as "policy," a set of one or more decisions of immediate interest are referred to as "strategy" or "strategic decisions" or just "decisions," and decisions that can be deferred until later are referred to as "tactics." The users confirm or modify **129** the policies, decisions and tactics. For example, the users may not want to address a particular decision at this time, in which case it would become a tactic.

Working in parallel with the first Framing module **314** is a second Framing module **316**. Such second Framing module **316** receives as input pertinent uncertainties or risks (uncontrollables), information sources and values that further describe the capabilities of the decision application **124**. The second Framing module **316** also receives as input the decisions identified by the first Framing module **314** and users' confirmation or modification **129** of the values, information sources and uncertainties. With such, the second Framing module **316** structures a relationship of decisions, values and uncertainties in form of the influence diagram and a corresponding directory to sources of information **306**.

FIG. **4** illustrates an example of Alternatives **400** in accordance with one embodiment of the present invention. The purpose of the Alternatives process is to develop a set of strategic alternatives that capture the range of possibilities envisioned by the users. After Framing, the platform moves to Alternatives, and receives from the decision application **124** alternative strategies each comprised of a set of coherent choices for each of the strategic decision. The users confirm or modify **129** the alternative strategies. The platform generates the visual display of the strategies defined on a strategy table **402**.

FIG. **4***a* illustrates various logic **406** associated with the Alternatives process of the present invention which is capable of generating several strategies defined on a strategy table **402**. Included with the Alternatives logic **406** is a first Alternatives module **410** that receives the decision hierarchy **304** generated by the Framing logic **310**. The first Alternatives module **410** obtains decision alternatives in each of the decision areas from the decision application **124** and from an information database **126** for the purpose of developing a strategy table. Each (strategic) decision from the decision hierarchy **304** becomes a column heading in the strategy table **402** with the alternatives for that decision arranged in a column beneath it. The first Alternatives module **410** also takes as input the users confirmation or modification **129** of the decision alternatives.

A second Alternatives module **412** combines the strategy table output of the first Alternatives module **410** with strategy descriptions from the decision application **124**. The strategy descriptions include a strategy name and the selection of one alternative for each of the decisions that comprise the column headings in the strategy table **402**. The second Alternatives module **412** can then display the strategies on a strategy table and incorporate the users' confirmation or modifications **129**. For example, the users may want to define their own strategy, which they would do by providing the second Alternatives module **412** with a strategy name and the selection of and alternative in each column of the strategy table **402**.

FIG. **5** illustrates an example of Analysis **500** in accordance with one embodiment of the present invention. The purpose of the Analysis process is to enable the users to have a shared understanding of the significant sources of risk and value in each of the initially defined alternative strategies. During Analysis, the platform prompts the information database **126** for assessments on each of the uncertainties set forth in a format **127** specified as low estimate, nominal estimate, and high estimate. These assessments are made for uncertainties influenced by the choice of decision, as well as independent uncertainties.

Using the information generated previously and the model structure of the decision application **124**, the platform makes the necessary calculations to output tornado diagrams **502** and decision sensitivity output displays for each of the alternative strategies **509**. The users confirm or modify the input information **129** and structure from the decision application **124**. The tornado diagrams identify the sources of significant risk in each alternative strategy and the decision sensitivity identifies the sources of significant value in each alternative strategy.

FIG. **5***a* illustrates various logic **506** associated with the Analysis process of the present invention. As shown, a first Analysis module **508** receives as input the influence diagram

306, identifying uncertainties and their relationship to the value and the decisions. The influence diagram also includes an information directory, which specifies the information databases **126** that will provide the decision-relevant information. This first Analysis module **508** also receives as input from the information data bases **126** assessed ranges or probabilities for each of the uncertainties identified by the influence diagram **306** generated using the Framing logic **310**. These data ranges are confirmed or modified by the users **129**.

The output of the first Analysis module **508** is further used by a second Analysis module **514**. The second Analysis module **514** takes as input the structural relationship of decisions, values and uncertainties from the decision application **124**. An example of such a structural relationship is a spreadsheet comprised of equations relating decisions, values and uncertainties. This output is, in turn, used to generate the tornado diagram **502** by varying each of the uncertainties over its range and recording the effect on value.

In parallel with the first and second Analysis modules is a third Analysis module **510** that takes as input the strategies defined on the strategy table **402**, the output of the first Analysis module **508** and the structural relationship of decisions, values and uncertainties from the decision application **124**. With such input, the third Analysis module **510** identifies a contribution to the total value of each alternative for each decision that comprises each strategy. Given this information, a decision sensitivity table **509** may be constructed.

FIG. **6** illustrates an example of Connection **600** in accordance with one embodiment of the present invention. The purpose of Connection is for the users to develop a new, more valuable "hybrid" strategy **602** combining the most valuable decisions in each of the initially defined alternative strategies. During Connection, the users' insight into the sources of risk and value **129** interacts with new decision relevant information from the database **126** and the decision structure provided by the decision application **124** to output an evaluation of the hybrid strategy **602**.

FIG. **6***a* illustrates various logic **604** associated with the Connection process of the present invention. As shown, the logic **604** includes a first Connection module **606** which receives as input a value contribution of each alternative for each decision that comprise each strategy, the decision sensitivity **509** generated by the Analysis logic **506**. The first connection module **606** also receives as input user insight **129** regarding how to combine the sources of value into a new, more valuable hybrid strategy. A second logic module **608** of the connection logic **604** takes as input the users' insight about additional information sources that could reduce the significant uncertainties or risks identified in the tornado diagram **502**. This second Connection module **608** then selects that new information from an appropriate decision relevant database (perhaps one not previously used for this decision problem) **126**. The description of the new hybrid alternative from the first Connection module **606** and the new risk reducing information from the second Connection module **608** are input to a third module **610**. This third module **610** uses the structural relationship of decisions, values and uncertainties (e.g., spreadsheet) from the decision application **124** to output the value of the hybrid strategy **602**.

FIG. **7** illustrates the various logical connectivity among the various common displays of the Framing, Alternatives, Analysis, and Connection that comprise the users' interface **128**.

FIGS. **8***a–i* illustrate an example of an application of the various logic components set forth in FIGS. **3**–**7**. As shown, such illustrative application of the collaborative decision platform relates to an individual and his/her spouse, the users, selecting a strategy for participation in an employer's stock purchase program. Initially, the collaborative decision platform executes a decision application selected by the users for developing stock purchase strategies.

In the Framing process, the collaborative decision platform uses input from the decision application to present the users with an initial decision hierarchy, which the users confirm or modify. The collaborative decision platform produces the resulting decision hierarchy **800**, shown in FIG. **8***a*, as an output, which identifies the decisions that are within the scope of the current decision making process.

The collaborative decision platform also uses input from the decision application to present the users with an initial influence diagram, which the users confirm or modify. The influence diagram identifies the critical uncertainties or risks, the decisions and the values that are important to the users, and it displays the relationships among them. The users confirm or modify the influence diagram. The collaborative decision platform produces the resulting influence diagram **802**, shown in FIG. **8***b*, as another output. Note that a directory of information sources **803** is included with the influence diagram.

The users are allowed to modify the influence diagram and the decision hierarchy only to the extent that the modifications were anticipated by the author of the application. This restriction assures that the alternative strategies that are defined in the Alternatives process can be analyzed with the spreadsheet provided by the decision application.

In the Alternatives process, the collaborative decision platform uses input from the decision application to present the users with an initial strategy table that is consistent with the decision hierarchy, which the users confirm or modify. One or more strategy names and their corresponding definitions on the strategy table are also presented to the users. The users may confirm or modify the strategies, including developing new strategies. The resulting strategy alternatives are displayed on strategy tables **804**, as shown in FIGS. **5***c* and **8***d*.

In the Analysis process, ranges on each uncertainty or risk **806** are input from the decision-relevant database specified on the influence diagram **802**, as shown in FIG. **8***e*. The users may confirm or modify the ranges. The collaborative decision platform takes as input the spreadsheet residing in the decision application that includes equations and data relating the decisions and uncertainties to the value, which in this case is profit. The collaborative decision platform uses the spreadsheet, strategies and uncertainty ranges to produce the tornado diagram **808** and decision sensitivity **810** shown in FIGS. **8***f* and **8***g*.

In the connection process, the users define on the strategy table **804** a new, more valuable "hybrid" strategy **811** that combines the most valuable alternatives from each of the initially defined alternative strategies, as shown in FIG. **8***h*. In defining this hybrid strategy, the users are relying heavily on the shared insight and understanding from the tornado diagram and decision sensitivity. The collaborative decision platform uses the spreadsheet from the decision application to calculate the value of the hybrid **812**, as shown on FIG. **8***i*.

FIG. **9** illustrates a method **900** for affording customer-centric collaborative decision-making in a business-to-business framework. In one embodiment, the method **900** may be carried using the collaborative decision platform set forth hereinabove. In the alternative, the present method may be executed using any other desired architecture.

Initially, in operation **902**, a minimum set of attributes is defined. Thereafter, first information regarding each of the minimum set of attributes is received from a receiving business. Note operation **904**. Second information is then received regarding proposed products or services in terms of the minimum set of attributes, as indicated in operation **906**. Such second information is received from a supplying business.

In use, a decision process is executed based on the first information and the second information as to which products or services is suitable for the receiving business. Note operation **908**. The present embodiment thus provides a customer-centric collaborative protocol that defines the minimum informational requirement for collaborative decision-making between enterprises (B2B).

The customer-centric collaborative protocol exploits a commonality in the attributes of the value structure of many enterprises that is sufficient to assess the implications of many decisions. An illustrative minimum set of attributes could include: price, sales, variable cost, fixed cost and investment. For many strategic decisions, knowing the affect of the decision on these attributes enables the enterprise to make an informed decision.

There are well-defined algorithms for the hierarchical expansion of each of the attributes in the minimum set in the event additional detail is required. When more detail is required, it may be nested within the higher level attributes. An expanded set of attributes could include: price, market share, market size, labor cost, material cost, administrative cost, annual expenses, working capital, plant and equipment, etc. The protocol or structure of the informational requirement is identical for a wide range of enterprises and many decisions within those enterprises, but the relative value of each attribute will be different. FIG. **9***a* illustrates a table **920** showing various customer-centric collaborative (C^3) attributes, and the value of a one-percent increases of such attributes in two different industries.

In accordance with the present invention, the supplying enterprise is required to describe its alternatives in terms of their effect on the value attributes that matter to the receiving enterprise. FIG. **10** illustrates a table **1000** showing such an effect on the value attributes.

FIG. **11** is a schematic diagram showing the customer-centric collaborative (C^3) protocol. As set forth hereinabove, the protocol defines the minimum informational requirement for decision making between enterprises (B2B). The value of improvements of each of the attributes is specified for a receiving enterprise **1100**. It should be noted that attributes are easily calculable for enterprises that focus on profit. However, even for enterprises that are not focused on profit, these same attributes are of critical importance. A supplying enterprise **1102** provides one or more alternative "attribute bundles" that describe products and services it is willing to deliver in terms of the attributes that matter to the receiving enterprise. An attribute bundle specifies how much of each attribute will be provided. It should be understood that the attribute levels can be assessed with little difficulty, using for example an influence diagram. A decision module **1104** may then execute the method **900** of FIG. **9**. FIG. **12** illustrates a first example **1200** of the embodiment set forth in FIG. **11**. As shown, an industry independent, open and scalable platform may be provided that uses the customer-centric collaborative protocol for real-time, remote collaborative decision making among enterprises. The customer-centric collaborative protocol can be used with an architecture or process that supports collaborative decision-making, such as a collaborative decision platform **1202** which is similar to that set forth hereinabove.

FIGS. **13** and **14** illustrate a second and third example **1300** and **1400** of the embodiment set forth in FIG. **11**. In the embodiment of FIG. **13**, the customer-centric collaborative protocol and an architecture or process that supports collaborative decision making, such as the collaborative decision platform, may together enable an open, scalable, industry independent process for real-time, remote decision-making between a receiving enterprise **1302** and a supplying enterprise **1304**. As shown, the present embodiment may serve to negotiate an agreement **1306** to purchase and deliver the highest value combination of attributes. In a third embodiment shown in FIG. **14**, the customer-centric collaborative protocol and an architecture or process that supports collaborative decision making, such as the collaborative decision platform, may together enable an open, scalable, industry independent process for real-time, remote decision-making among a receiving enterprise **1402** and supplying enterprises **1404**. As shown, the present embodiment may serve to negotiate an agreement **1406** to purchase and deliver the highest value combination of attributes.

FIG. **15** illustrates a fourth examples **1500** of the embodiment set forth in FIG. **11**, where an industry independent, open and scalable platform is provided for B2B exchange of existing goods and services that are not commodities. In other words, an effective platform for a non-commodity exchange is afforded.

As shown in FIG. **15**, the alternative attribute bundle can be offered by different enterprises **1504** to a receiving enterprise **1502** and need not be commodities, but rather may differ on the level offered of every attribute. It should be understood that commodities are goods and services that can be defined without the information about or the interaction of the customer. As shown in FIG. **15**, the customer-centric collaborative protocol and an architecture or process that supports collaborative decision making, such as the collaborative decision platform, together enable an industry-independent, open and scalable platform for the real-time B2Bexchange of existing goods and services **1502** that are not commodities.

FIG. **16** illustrates a fifth example **1600** of the embodiment set forth in FIG. **11**, where an industry independent, open and scalable platform is provided for B2Breal-time collaboration in the definition of future, non-existent goods and services **1601**. As shown in FIG. **16**, the alternative attribute bundles can be offered by different enterprises and need not exist. Rather, they may represent proposals to deliver goods and services that could be developed in the future. As shown, an agreement **1606** may be negotiated to deliver the highest value combination of attributes in the future. FIGS. **17** and **18** illustrate sixth and seventh examples **1700** and **1800**, respectively, of the embodiment set forth in FIG. **11**, where a new business design is provided that assists B2B enterprises in measuring the value creation for its customers.

As shown in FIG. **17**, the customer-centric collaborative protocol and publicly available information **1702** may together enable a new business design that assists B2B enterprises in measuring the prospective value creation for its customers. With reference to FIG. **18**, a particular embodiment of that business design could include the customer-centric collaborative protocol, publicly available information **1702** and a collaborative decision platform **1802**, which together enable a new business design that assists B2B enterprises in measuring the retrospective value creation for its customers.

An exemplary application of a customer-centric collaborative protocol utilizing the collaborative decision platform for the selection of a strategy for "Customer Relationship Management (CRM)" will now be set forth. In particular, the present B2B example relates to a receiving enterprise desirous of an improved CRM strategy and a supplying enterprise capable of delivering alternative CRM strategies.

In this case during the Framing process, the receiving enterprise provides the policies, which constrain the strategic alternatives. The supplying enterprise demonstrates its experience by offering a list of strategic decisions. The receiving enterprise believes that two of the decisions are tactical, i.e. can be made later. FIG. **19** illustrates the resulting decision hierarchy **1900** developed collaboratively and asynchronously. FIG. **20** shows the influence diagram **2000**, which identifies the critical uncertainties, the strategic decisions and the attributes **2020** that are of value to the receiving enterprise and which display the relationship among them. For two of the attributes, more detail is required and the higher level attributes are expanded hierarchically in those areas **2100** and **2200**, as shown in FIGS. **21** and **22**, respectively.

During the Alternatives process, three alternative strategies **2300**, **2302**, and **2304** are defined collaboratively on a strategy table in terms of the strategic decisions, as shown in FIGS. **23***a*, **23***b* and **23***c*, respectively. The strategy table is developed remotely and asynchronously. The strategies are developed in the physical presence of both enterprises.

In the Analysis process, the supplying enterprise uses information from its database to assess the range of effect that the "Revenue Growth" strategy will have on each of the attributes **2410**. Note **2400** in FIG. **24**. Similar assessments are made for each of the other strategies. The receiving enterprise may establish its value for changes in each of the attributes as shown in the table **2500** of FIG. **25**.

The table **2600** in FIG. **26** shows the calculations performed inside the collaborative decision platform when the customer-centric collaborative protocol is used. As shown, the value of an alternative to the client can be estimated by multiplying the improvement in each attribute by the customer's value for changes in that attribute.

The remarkable simplicity of these calculations enables shared insight into the sources of risk and sources of value, which is displayed in the tornado diagram **2700** and decision sensitivity **2800** for each of the alternative strategies, as shown in FIGS. **27** and **28**, respectively. It should be noted that different solutions might be appropriate for clients in different industries because of different client values for the C^3 attributes.

Using the shared understanding of the sources of risk and value in the initially defined alternative strategies, the supplying and receiving enterprise collaborate in developing a new, more valuable "hybrid" strategy **2900**, as shown in FIG. **29**. Its corresponding decision sensitivity **3000** of FIG. **30** compares the total value of the hybrid strategy with the initially defined alternatives and identifies its sources of value.

While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of a preferred embodiment should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A method for providing customer-centric collaborative decision making in a business-to-business framework, comprising the steps of:

(a) defining a minimum set of attributes;

(b) receiving first information regarding each of the minimum set of attributes from a receiving business;

(c) receiving second information regarding proposed products or services in terms of the minimum set of attributes, wherein the second information is received from a supplying business;

(d) executing a decision process based on the first information and the second information, where the decision process determines as to which products or services is suitable for the receiving business;

(e) wherein the steps are carried out using a system capable of:

(i) executing an application capable of performing decision logic;

(ii) retrieving information from a database in accordance with the decision logic;

(iii) receiving information from a user in accordance with the decision logic utilizing a user interface;

(iv) processing the information utilizing the decision logic; and

(v) wherein (i)–(iv) are carried out by a collaborative decision platform capable of accomplishing (ii)–(iv) for different purposes by executing different applications each capable of performing different decision logic;

(f) wherein an application interface provides an interface between the application and the collaborative decision platform, where (ii)–(iv) are carried out using universal modules capable of interfacing with different applications adapted for applying the universal modules to different business sectors;

(g) wherein the collaborative decision platform communicates with the application through a standard interface protocol.

2. The method as recited in claim **1**, wherein the attributes include price, sales, variable costs, fixed cost, and investment.

3. The method as recited in claim **2**, wherein the attributes further include market share, market size, labor cost, material cost, administrative cost, annual expenses, working capital, planning and equipment.

4. The method as recited in claim **1**, wherein the first information and second information are received utilizing a network.

5. The method as recited in claim **1**, wherein the network is the Internet.

6. The method as recited in claim **1**, wherein the business sectors include real estate, medicine, corporate, and financial.

7. The method as recited in claim **1**, wherein:

data is collected including (i) policies that form boundary conditions associated with the decision logic, (ii) strategic decisions to be made, (iii) values that are important to the user, (iv) uncertainties that impact the values, and a relationship between (i)–(iv);

a strategy table is created using the data, where each column heading in the strategy table includes a strategic decision from a decision hierarchy with alternatives for a decision arranged therebeneath;

the uncertainties are assessed for analysis purposes;

a tornado diagram and decision sensitivity output displays are generated, where the tornado diagram identifies sources of significant risk in each of a plurality of alternative strategies and the decision sensitivity output displays identify sources of significant value in each of the alternative strategies; and

the data is collected from decision logic for generating visual displays of a decision hierarchy and an influence diagram, where the user is prompted to approve the visual displays of the decision hierarchy and the influence diagram.

8. A computer program product for providing customer-centric collaborative decision making in a business-to-business framework, comprising:

(a) computer code for defining a minimum set of attributes;

(b) computer code for receiving first information regarding each of the minimum set of attributes from a receiving business;

(c) computer code for receiving second information regarding proposed products or services in terms of the minimum set of attributes, wherein the second information is received from a supplying business;

(d) computer code for executing a decision process based on the first information and the second information, where the decision process determines as to which products or services is suitable for the receiving business;

(e) wherein the computer code is executed using a system capable of:

(i) executing an application capable of performing decision logic;

(ii) retrieving information from a database in accordance with the decision logic;

(iii) receiving information from a user in accordance with the decision logic utilizing a user interface;

(iv) processing the information utilizing the decision logic; and

(vi) wherein (i)–(iv) are carried out by a collaborative decision platform capable of accomplishing (ii)–(iv) for different purposes by executing different applications each capable of performing different decision logic;

(f) wherein an application interface provides an interface between the application and the collaborative decision platform where (ii)–(iv) are carried out using universal modules capable of interfacing with different applications adapted for applying the universal modules to different business sectors;

(g) wherein the collaborative decision platform communicates with the application through a standard interface protocol.

9. The computer program product as recited in claim **8**, wherein the attributes include price, sales, variable costs, fixed cost, and investment.

10. The computer program product as recited in claim **9**, wherein the attributes further include market share, market size, labor cost, material cost, administrative cost, annual expenses, working capital, planning and equipment.

11. The computer program product as recited in claim **8**, wherein the first information and second information are received utilizing a network.

12. The computer program product as recited in claim **8**, wherein the network is the Internet.

13. A system for providing customer-centric collaborative decision making in a business-to-business framework, comprising:

(a) logic for defining a minimum set of attributes;

(b) logic for receiving first information regarding each of the minimum set of attributes from a receiving business;

(c) logic for receiving second information regarding proposed products or services in terms of the minimum set of attributes, wherein the second information is received from a supplying business;

(d) logic for executing a decision process based on the first information and the second information, where the decision process determines as to which products or services is suitable for the receiving business;

(e) wherein the system is further capable of:

(i) executing an application capable of performing decision logic;

(ii) retrieving information from a database in accordance with the decision logic;

(iii) receiving information from a user in accordance with the decision logic utilizing a user interface;

(iv) processing the information utilizing the decision logic; and

(v) wherein (i)–(iv) are carried out by a collaborative decision platform capable of accomplishing (ii)–(iv) for different purposes by executing different applications each capable of performing different decision logic;

(f) wherein an application interface provides an interface between the application and the collaborative decision platform, where (ii)–(iv) are carried out using universal modules capable of interfacing with different applications adapted for applying the universal modules to different business sectors;

(g) wherein the collaborative decision platform communicates with the application through a standard interface protocol.

14. The system as recited in claim **13**, wherein the attributes include price, sales, variable costs, fixed cost, and investment.

15. The system as recited in claim **14**, wherein the attributes further include market share, market size, labor cost, material cost, administrative cost, annual expenses, working capital, planning and equipment.

16. The system as recited in claim **13**, wherein the first information and second information are received utilizing a network.

17. The system as recited in claim **13**, wherein the network is the Internet.

* * * * *